

Міністерство освіти і науки України
ЗНТУ

Кафедра програмних засобів

Архітектура комп'ютера

Конспект лекцій
для студентів спеціальності

121 Інженерія програмного забезпечення
усіх форм навчання

Введення в дисципліну.

Комп'ютер виконує чотири інформаційні процеси: введення, опрацювання, виведення і зберігання даних. Усі сучасні комп'ютери, незважаючи на різноманіття моделей, мають основні складові, що забезпечують виконання цих процесів.

Архітектура ПК — опис сукупності пристроїв та блоків ПК і зв'язків між ними.

Поняття архітектури тісно пов'язане з принципами роботи ПК. Архітектура визначає принципи дії, інформаційні зв'язки і взаємодію основних складових ПК: процесора, внутрішньої і зовнішньої пам'яті та периферійних пристроїв. Уніфікація архітектури ПК забезпечує їх сумісність з точки зору користувача.

Принципи функціонування сучасних комп'ютерів

1. *Принцип двійкового кодування полягає у тому, що всі дані подають у вигляді двійкових кодів.*
2. *Принцип програмного керування полягає у тому, що всі операції з опрацюванням даних здійснюють відповідно до програм і ці програми розташовують у пам'яті комп'ютера.*
3. *Принцип адресності полягає у такій організації пам'яті комп'ютера, за якої процесор може безпосередньо звернутись до даних, розташованих у будь-якій частині пам'яті. А кожна найменша частина пам'яті (комірка пам'яті) має унікальну назву – адресу.*
4. *Принцип однорідності пам'яті полягає у тому, що всі дані, у тому числі й програми, зберігають в одному і тому самому пристрої пам'яті.*

З часом принципи побудови комп'ютера розвивали, їм надавали нового змісту, їх доповнювали. Під час розробки комп'ютерів у кінці 70-х років ХХ століття й особливо під час створення першого персонального комп'ютера корпорації ІВМ (ІВМ РС, 1981 р.), було сформульовано таке положення.

Магістрально-модульний принцип

1. *Дані між окремими пристроями комп'ютера передають єдиною магістраллю — системною шиною, в якій виділяють три окремі шини: шину даних, шину вказівок і шину адрес;*
2. *Комп'ютер складається з окремих блоків — модулів, кожний з яких виконує певні функції.*

Останнє дає змогу звести модернізацію або ремонт комп'ютера до заміни окремих модулів. Так, можна замінити процесор, блоки пам'яті, монітор на аналогічні або на пристрої з покращеними значеннями властивостей.

Виконання програми починають з того, що пристрій керування зчитує пам'ять комірки, у якій міститься перша вказівка програми, та організовує її виконання. Команда надходить до АЛП, який виконує певну операцію. Після виконання однієї вказівки пристрій управління (ПУ) починає виконання вказівки з наступної комірки пам'яті. Порядок комірок операційної системи, з яких відбувалося зчитування, визначається за допомогою вказівок ПУ. Тобто ПУ виконує програми автоматично, без втручання людини, в чому й полягає принцип програмного керування.

Вказівка (команда) — це опис (запис) (елементарної) операції, яку повинен виконати комп'ютер.

В загальному випадку, вказівка містить таку інформацію:

- код операції, яку потрібно виконати;
- приписи для визначення операндів чи їх адреси;
- приписи для розміщення одержаного результату.

Центральний процесор розуміє і виконує лише *визначену* систему команд: ті коди, які вказують на виконання певних операцій. Будь-яка програму подають у вигляді послідовності таких кодів. Під час роботи програми центральний процесор в будь-який момент виконує одну з вказівок з дуже великою швидкістю. Також операційна система надає користувачу певний набір вказівок і відповідні їм відповіді, за допомогою яких можна керувати ЕОМ.

Введення

Клавіатура і миша — найпоширеніші пристрої для взаємодії з комп'ютером. Програмне забезпечення, що керує машиною, перетворює комбінації введених символів або як вказівки, які можна і потрібно виконати, або як дані, що підлягають опрацюванню. Прості програми можна ввести безпосередньо з клавіатури, але складніші й більші програми зазвичай завантажують в оперативну пам'ять за допомогою спеціального зовнішнього пристрою, яке передає до машини збережену на диску інформацію.

Опрацювання

У блоці *центрального процесора пристрій керування* пильнує за порядком виконання операцій. *Арифметико-логічний пристрій* виконує арифметичні і логічні операції. Активну програму розміщують в оперативній пам'яті комп'ютера, завдяки чому центральний процесор може вибирати вказівки послідовно, одну за одною. Програми, які завжди зберігають у сталій пам'яті, забезпечують початкову активацію комп'ютера і підтримують взаємодію процесора с пристроями введення-виведення.

Виведення

Монітор забезпечує відображення результатів опрацювання інформації. Зазвичай комп'ютер виводить на екран інформацію, яка надходить з клавіатури, а також свої відповіді. *Принтер* слугує для виведення інформації на папір. Інформацію можна виводити і електричними імпульсами.



В кінці 70-х років ХХ століття В. М. Глушков запропонував принцип макроконвеєрної архітектури ЕОМ з багатьма потоками команд і даних. За сучасною класифікацією — архітектура МІМД (англійською multiple instruction, multiple data — множинний потік команд, множинний потік даних). Розробку макроконвеєрної ЕОМ було виконано в Інституті

кібернетики АН України під керівництвом В.М. Глушкова С.Б. Погребинським (головний конструктор), В.С. Михалевичем, О.А. Летичевським, І.М. Молчановим. Машина ЕС–2701 у 1984 р. і обчислювальна система ЕС–1766 у 1987 р. були передані в серійне виробництво на Пензенський завод ЕОМ. На той період це були найпотужніші в СРСР обчислювальні системи з номінальною потужністю понад 10^9 оп./сек. При цьому у багатопроцесорній системі забезпечено майже лінійне зростання продуктивності під час нарощування обчислювальних ресурсів і динамічна реконфігурація. Вони не мали аналогів у світовій практиці і стали оригінальним напрямком розвитку високопродуктивних систем.

Що таке мікропроцесор?

Ядром будь-якої мікропроцесорної системи є мікропроцесор або просто процесор (від англійського processor). Перекласти на українську мову це слово перекладається як «обробник», оскільки саме мікропроцесор - це той вузол, блок, який проводить всю обробку інформації усередині мікропроцесорної системи. Решта вузлів виконує всього лише допоміжні функції: зберігання інформації (у тому числі і управляючої інформації, тобто програми), зв'язки із зовнішніми пристроями, зв'язки з користувачем і т.д. Процесор замінює практично всю «жорстку логіку», яка знадобилася б у разі традиційної цифрової системи. Він виконує арифметичні функції (додавання, множення і т.д.), логічні функції (зсув, порівняння, маскування кодів і т.д.), тимчасове зберігання кодів (у внутрішніх регістрах), пересилку кодів між вузлами мікропроцесорної системи і інше. Кількість таких елементарних операцій, виконуваних процесором, може досягати декількох сотень. Процесор можна порівняти з мозком системи. Але при цьому треба враховувати, що всі свої операції процесор виконує послідовно, тобто одну операцію за іншою, по черзі. Звичайно, існують процесори з паралельним виконанням деяких операцій, зустрічаються також мікропроцесорні системи, в яких декілька процесорів працюють над однією задачею паралельно, але це не часто. З одного боку, послідовне виконання операцій - безперечна перевага, оскільки дозволяє за допомогою всього лише одного процесора виконувати будь-які найскладніші алгоритми обробки інформації. Але, з другого боку, послідовне виконання операцій призводить до того, що час виконання алгоритму залежить від його складності. Прості алгоритми виконуються швидше складних. Тобто мікропроцесорна система здатна зробити все, але працює вона не дуже швидко, адже всі інформаційні потоки доводиться пропускати через один-єдиний вузол - мікропроцесор (мал. 1.3). В традиційній цифровій системі можна легко організувати паралельну обробку всіх потоків інформації, правда, ціною ускладнення схеми.

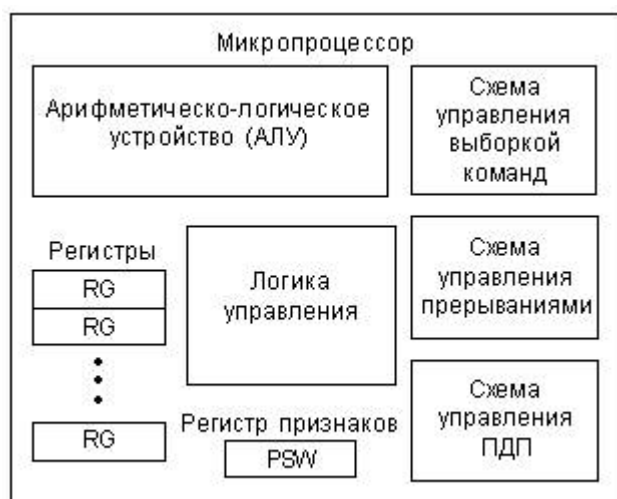


Мал. 1.3. Інформаційні потоки в мікропроцесорній системі.

Отже, мікропроцесор здатний виконувати безліч операцій. Але звідки він знає, яку операцію йому треба виконувати в даний момент? Саме це визначається управляючою інформацією, програмою. Програма є набором команд (інструкцій), тобто цифрових кодів, розшифрувавши які, процесор знає, що йому треба робити. Програма від початку і до кінця складається людиною, програмістом, а процесор виступає в ролі слухняного виконавця цієї програми, ніякої ініціативи він не проявляє (якщо, звичайно, справний). Тому порівняння процесора з мозком не дуже коректне. Він всього лише виконавець того алгоритму, який наперед склала для нього людина. Будь-яке відхилення від цього алгоритму може бути викликано тільки несправністю процесора або яких-небудь інших вузлів мікропроцесорної системи.

Всі команди, виконувані процесором, утворюють систему команд процесора. Структура і об'єм системи команд процесора визначають його швидкість, гнучкість, зручність використання. Всього команд у процесора може бути від декількох десятків до декількох сотень. Система команд може бути розрахована на вузький круг вирішуваних задач (у спеціалізованих процесорах) або на максимально широкий круг задач (в універсальних процесорах). Коди команд можуть мати різну кількість розрядів (займати від одного до декількох байт). Кожна команда має свій час виконання, тому час виконання всієї програми залежить не тільки від кількості команд в програмі, але і від того, які саме команди використовуються.

Для виконання команд в структуру процесора входять внутрішні регістри, арифметико-логічний пристрій (АЛП, ALU - Arithmetic Logic Unit), мультиплексори, буфери, регістри і інші вузли. Робота всіх вузлів синхронізується загальним зовнішнім тактовим сигналом процесора. Тобто процесор є досить складним цифровим пристроєм (мал. 1.4).



Мал. 1.4. Приклад структури найпростішого процесора.

Втім, для розробника мікропроцесорних систем інформація про тонкість внутрішньої структури процесора не дуже важлива. Розробник повинен розглядати процесор як «чорний ящик», який у відповідь на вхідні і управляючі коди проводить ту або іншу операцію і видає вихідні сигнали. Розробнику необхідно знати систему команд, режими роботи процесора, а також правила взаємодії процесора із зовнішнім світом або, як їх ще називають, протоколи обміну інформацією. Про внутрішню структуру процесора треба знати тільки те, що необхідне для вибору тієї або іншої команди, того або іншого режиму роботи.

Архітектура мікропроцесорних систем

Дотепер ми розглядали тільки один тип архітектури мікропроцесорних систем - архітектуру із загальною, єдиною шиною для даних і команд (одношинну, або прінстонську, фон-

неймановську архітектуру). Відповідно, у складі системи в цьому випадку присутня одна загальна пам'ять, як для даних, так і для команд (мал. 1.15).



Мал. 1.14. Архітектура із загальною шиною даних і команд.

Але існує також і альтернативний тип архітектури мікропроцесорної системи - це архітектура з роздільними шинами даних і команд (двохшинна, або гарвардська, архітектура). Ця архітектура припускає наявність в системі окремої пам'яті для даних і окремої пам'яті для команд (мал. 1.16). Обмін процесора з кожним з двох типів пам'яті відбувається по своїй шині. Архітектура із загальною шиною поширена набагато більше, вона застосовується, наприклад, в персональних комп'ютерах і в складних мікрокомп'ютерах. Архітектура з роздільними шинами застосовується в основному в однокристальних мікроконтролерах.

Розглянемо деякі достоїнства і недоліки обох архітектурних рішень. Архітектура із загальною шиною (прінстонська, фон-неймановська) простіша, вона не вимагає від процесора одночасного обслуговування двох шин, контролю обміну по двох шинах відразу. Наявність єдиної пам'яті даних і команд дозволяє гнучко розподіляти її об'єм між кодами даних і команд. Наприклад, в деяких випадках потрібна велика і складна програма, а даних в пам'яті треба зберігати дуже не багато. В інших випадках, навпаки, програма потрібна проста, але необхідні великі об'єми бережених даних. Перерозподіл пам'яті не викликає жодних проблем, головне - щоб програма і дані разом поміщалися в пам'яті системи. Як правило, в системах з такою архітектурою пам'ять буває досить великого об'єму (до десятків і сотень мегабайт). Це дозволяє вирішувати найскладніші задачі.



Мал. 1.15. Архітектура з роздільними шинами даних і команд.

Архітектура з роздільними шинами даних і команд складніша, вона примушує процесор працювати одночасно з двома потоками кодів, обслуговувати обмін по двох шинах одночасно. Програма може розміщуватися тільки в пам'яті команд, дані - тільки в пам'яті

даних. Така вузька спеціалізація обмежує круг задач, вирішуваних системою, оскільки не дає можливості гнучкого перерозподілу пам'яті. Пам'ять даних і пам'ять команд в цьому випадку мають не дуже великий об'єм, тому використання систем з даною архітектурою обмежується звичайно не дуже складними задачами. В чому ж перевага архітектури з двома шинами (гарвардської)? В першу чергу, в швидкодії. Річ у тому, що при єдиній шині команд і даних процесор вимушений по одній цій шині приймати дані (з пам'яті або пристрою вводу/виводу) і передавати дані (в пам'ять або в пристрій вводу/виводу), а також читати команди з пам'яті. Природно, одночасно ці пересилки кодів по магістралі відбуватися не можуть, вони повинні проводитися по черзі. Сучасні процесори здатні сумістити в часі виконання команд і проведення циклів обміну по системній шині. Використовування конвейсних технологій і швидкої кеш-пам'яті дозволяє їм прискорити процес взаємодії з порівняно повільною системною пам'яттю. Підвищення тактової частоти і вдосконалення структури процесорів дають можливість скоротити час виконання команд. Але подальше збільшення швидкодії системи можливе тільки при поєднанні пересилки даних і читання команд, тобто при переході до архітектури з двома шинами.

У разі двохшинної архітектури обмін по обох шинах може бути незалежним, паралельним в часі. Відповідно, структури шин (кількість розрядів коду адреси і коду даних, порядок і швидкість обміну інформацією і т.д.) можуть бути вибрані оптимально для тієї задачі, яка розв'язується кожною шиною. Тому за інших рівних умов перехід на двохшинну архітектуру прискорює роботу мікропроцесорної системи, хоча і вимагає додаткових витрат на апаратуру, ускладнення структури процесора. Пам'ять даних в цьому випадку має свій розподіл адрес, а пам'ять команд - своє. Простіше за все переваги двохшинної архітектури реалізуються усередині однієї мікросхеми. В цьому випадку можна також істотно зменшити вплив недоліків цієї архітектури. Тому основне її використання - в мікроконтролерах, від яких не вимагається рішення дуже складних задач, але необхідна максимальна швидкодія при заданій тактовій частоті.

Адресація операндів

Велика частина команд процесора працює з кодами даних (операндами). Одні команди вимагають вхідних операндів (одного або двох), інші видають вихідні операнди (частіше один операнд). Вхідні операнди називаються ще операндами-джерелами, а вихідні називаються операндами-приймачами. Всі ці коди операндів (вхідні і вихідні) повинні десь розташовуватися. Вони можуть знаходитися у внутрішніх регістрах процесора (найзручніший і швидкий варіант). Вони можуть розташовуватися в системній пам'яті (найпоширеніший варіант). Нарешті, вони можуть знаходитися в пристроях вводу/виводу (самий окремих випадок). Визначення місця положення операндів проводиться кодом команди. Причому існують різні методи, за допомогою яких код команди може визначити, звідки брати вхідний операнд і куди поміщати вихідний операнд. Ці методи називаються методами адресації. Ефективність вибраних методів адресації багато в чому визначає ефективність роботи всього процесора в цілому.

Методи адресації

Кількість методів адресації в різних процесорах може бути від 4 до 16. Розглянемо декілька типових методів адресації операндів, що використовуються зараз в більшості мікропроцесорів.

Безпосередня адресація (мал. 3.1) припускає, що операнд (вхідний) знаходиться в пам'яті безпосередньо за кодом команди. Операнд звичайно є константою, яку треба кудись переслати, до чогось додати і т.д. Наприклад, команда може полягати в тому, щоб додати число 6 до вмісту якогось внутрішнього регістра процесора. Це число 6

розташовуватиметься в пам'яті, усередині програми в адресі, наступній за кодом даної команди складання.



Мал. 3.1. Безпосередня адресація.

Пряма (вона ж абсолютна) адресація (мал. 3.2) припускає, що операнд (вхідний або вихідний) знаходиться в пам'яті за адресою, код якого знаходиться усередині програми зразу ж за кодом команди. Наприклад, команда може полягати в тому, щоб очистити (зробити нульовим) вміст елемента пам'яті з адресою 1000000. Код цієї адреси 1000000 розташовуватиметься в пам'яті, усередині програми в наступній адресі за кодом даної команди очищення.



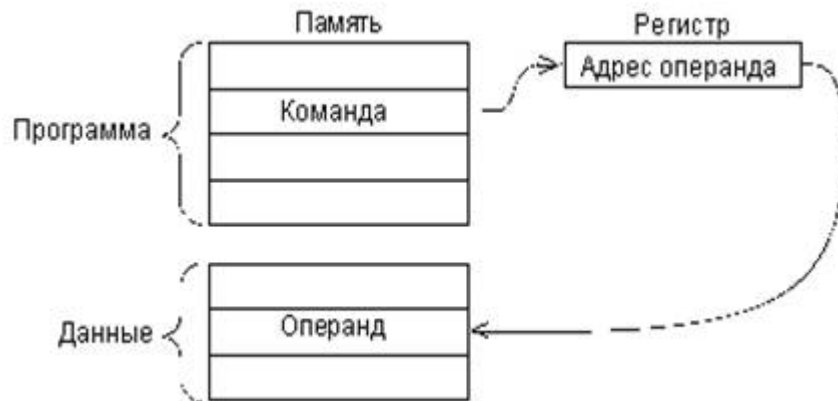
Мал. 3.2. Пряма адресація.

Регістрова адресація (мал. 3.3) припускає, що операнд (вхідний або вихідний) знаходиться у внутрішньому регістрі процесора. Наприклад, команда може полягати в тому, щоб переслати число з нульового регістра в перший. Номери обох регістрів (0 і 1) визначатимуться кодом команди пересилки.

Непрямо-регістрова (вона ж непряма) адресація припускає, що у внутрішньому регістрі процесора знаходиться не сам операнд, а його адреса в пам'яті (мал. 3.4). Наприклад, команда може полягати в тому, щоб очистити елемент пам'яті з адресою, що знаходиться в нульовому регістрі. Номер цього регістра (0) визначатиметься кодом команди очищення.



Мал. 3.3. Регістрова адресація.



Мал. 3.4. Непряма адресація.

Рідше зустрічаються ще два методи адресації. Автоінкрементна адресація дуже близька до непрямої адресації, але відрізняється від неї тим, що після виконання команди вміст реєстра, що використовується, збільшується на одиницю або на два. Цей метод адресації дуже зручний, наприклад, при послідовній обробці кодів з масиву даних, що знаходиться в пам'яті. Після обробки якогось коду адреса в реєстрі вказуватиме вже на наступний код з масиву. При використуванні непрямої адресації в даному випадку довелося б збільшувати вміст цього реєстра окремою командою. Автодекрементна адресація працює схоже на автоінкрементну, але тільки вміст вибраного реєстра зменшується на одиницю або на два перед виконанням команди. Ця адресація також зручна при обробці масивів даних. Сумісне використання автоінкрементної і автодекрементної адресацій дозволяє організувати пам'ять стекового типу (див. розділ 2.4.2). З інших поширених методів адресації можна згадати про індексні методи, які припускають для обчислення адреси операнда надбавку до вмісту реєстра заданої константи (індексу). Код цієї константи розташовується в пам'яті безпосередньо за кодом команди. Відзначимо, що вибір того або іншого методу адресації в значній мірі визначає час виконання команди. Найшвидша адресація - це реєстрова, оскільки вона не вимагає додаткових циклів обміну по магістралі. Якщо ж адресація вимагає звернення до пам'яті, то час виконання команди збільшуватиметься за рахунок тривалості необхідних циклів звернення до пам'яті. Зрозуміло, що чим більше внутрішніх реєстрів у процесора, тим частіше і вільніше можна застосовувати реєстрову адресацію, і тим швидше працювати система в цілому.

Сегментація пам'яті

Кажучи про адресацію, не можна обійти питання про сегментацію пам'яті, вживаної в деяких процесорах, наприклад в процесорах IBM PC-сумісних персональних комп'ютерів.

В процесорі Intel 8086 сегментація пам'яті організована таким чином. Вся пам'ять системи представляється не у вигляді безперервного простору, а у вигляді декількох областей - сегментів заданого розміру (по 64 Кбайта), положення яких в просторі пам'яті можна змінювати програмним шляхом. Для зберігання кодів адрес пам'яті використовуються не окремі регістри, а пари регістрів:

- сегментний регістр визначає адресу початку сегменту (тобто положення сегменту в пам'яті);
- регістр покажчика (регістр зсуву) визначає положення робочої адреси усередині сегменту.

При цьому фізична 20-розрядна адреса пам'яті, що виставляється на зовнішню шину адреси, утворюється так, як показано на мал. 3.5, тобто шляхом складання зсуву і адреси сегменту із зсувом на 4 біти. Положення цієї адреси в пам'яті показано на мал. 3.6. Сегмент може починатися тільки на 16-байтній межі пам'яті (оскільки адреса початку сегменту, по суті, має чотири молодші нульові розряди, як видно з мал. 3.5), тобто з адреси, кратної 16. Ці допустимі межі сегментів називаються межами параграфів. Відзначимо, що введення сегментації, перш за все, зв'язано з тим, що внутрішні регістри процесора 16-розрядні, а фізична адреса пам'яті 20-розрядна (16-розрядна адреса дозволяє використовувати пам'ять тільки в 64 Кбайт, що явно недостатньо). В процесорі MC68000 фірми Motorola, що з'явився в той же час, внутрішні регістри 32-розрядні, тому там проблеми сегментації пам'яті не виникає.

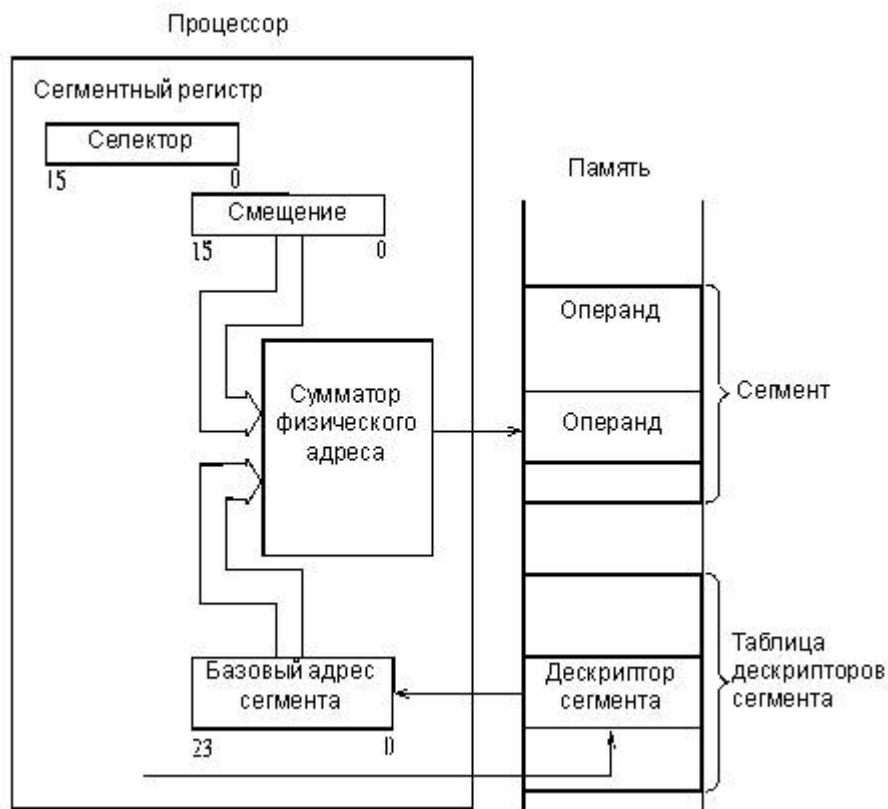


Мал. 3.5. Формування фізичної адреси пам'яті з адреси сегменту і зсуву.



Мал. 3.6. Фізична адреса в сегменті (всі коди - шістнадцяткові).

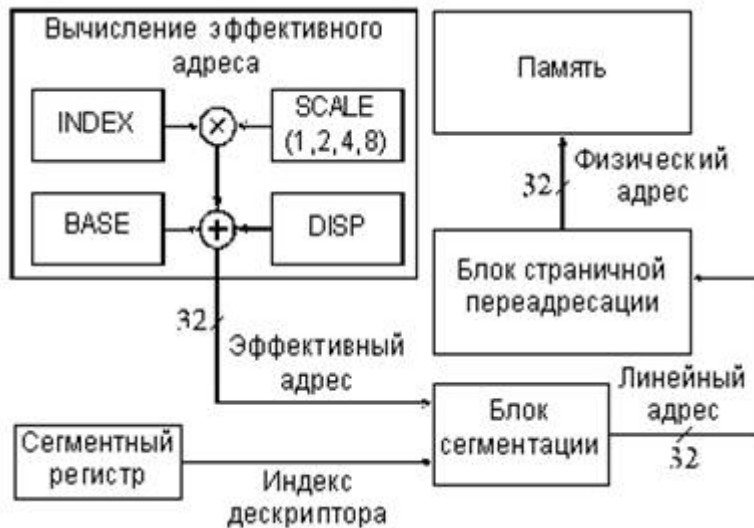
Застосовуються і складніші методи сегментації пам'яті. Наприклад, в процесорі Intel 80286 в так званому захищеному режимі адреса пам'яті обчислюється відповідно до мал. 3.7. В сегментному реєстрі в даному випадку зберігається не базова (початкова) адреса сегментів, а коди селекторів, що визначають адреси в пам'яті, по яких зберігаються дескриптори (тобто описувачі) сегментів. Область пам'яті з дескрипторами називається таблицею дескрипторів. Кожний дескриптор сегменту містить базову адресу сегменту, розмір сегменту (від 1 до 64 Кбайт) і його атрибути. Базова адреса сегменту має розрядність 24 біт, що забезпечує адресацію 16 Мбайт фізичної пам'яті.



Мал. 3.7. Адресация пам'яті в захищеному режимі процесора Intel 80286.

Таким чином, на суматор, що обчислює фізичну адресу пам'яті, подається не вміст сегментного реєстра, як у попередньому випадку, а базова адреса сегменту з таблиці дескрипторів.

Ще складніший метод адресації пам'яті з сегментацією використаний в процесорі Intel 80386 і в більш пізніх моделях процесорів фірми Intel. Цей метод ілюструється мал. 3.8. Адреса пам'яті (фізична адреса) обчислюється в три етапи. Спочатку обчислюється так звана ефективна адреса (32-розрядний) шляхом підсумовування трьох компонентів: бази, індексу і зсуву (Base, Index, Displacement), причому можливе множення індексу на масштаб (Scale). Ці компоненти мають наступний сенс:



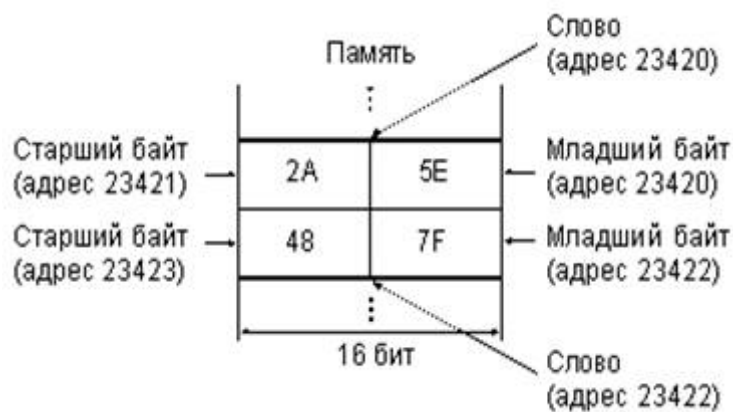
Мал. 3.8. Формування фізичної адреси пам'яті процесора 80386 в захищеному режимі.

- зсув - це 8-, 16- або 32-розрядне число, включене в команду.
- база - цей вміст базового реєстра процесора. Звичайно воно використовується для вказівки на початок деякого масиву.
- індекс - цей вміст індексного реєстра процесора. Звичайно воно використовується для вибору одного з елементів масиву.
- масштаб - це множник (він може бути рівний 1, 2, 4 або 8), вказаний в коді команди, на який перед підсумовуванням з іншими компонентами множиться індекс. Він використовується для вказівки розміру елемента масиву.

Потім спеціальний блок сегментації обчислює 32-розрядну лінійну адресу, яка є сумою базової адреси сегменту з сегментного реєстра з ефективною адресою. Нарешті, фізична 32-бітова адреса пам'яті утворюється шляхом перетворення лінійної адреси блоком сторінкової переадресації, який здійснює переклад лінійної адреси у фізичний сторінками по 4 Кбайта. У будь-якому випадку сегментація дозволяє виділити в пам'яті один або декілька сегментів для даних і один або декілька сегментів для програм. Перехід від одного сегменту до іншого зводиться всього лише до зміни вмісту сегментного реєстра. Іноді це дуже зручно. Але для програміста працювати з сегментованою пам'яттю звичайно складніше, ніж з безперервною, несегментованою пам'яттю, оскільки доводиться стежити за межами сегментів, за їх описом, перемиканням і т.д.

Адресація байт і слів

Багато процесорів, що мають розрядність 16 або 32, здатні адресувати не тільки ціле слово в пам'яті (16-розрядне або 32-розрядне), але і окремі байти. Кожному байту в кожному слові при цьому відводиться своя адреса. Так, у разі 16-розрядних процесорів всі слова в пам'яті (16-розрядні) мають парні адреси. А байти, що входять в ці слова, можуть мати як парні адреси, так і непарні. Наприклад, нехай 16-розрядний елемент пам'яті має адресу 23420, і в ній зберігається код 2A5E (мал. 3.9).



Мал. 3.9. Адресація слів і байтів.

При зверненні до цілого слова (з вмістом 2A5E) процесор виставляє адресу 23420. При зверненні до молодшого байта цієї комірки (з вмістом 5E) процесор виставляє ту ж саму адресу 23420, але використовує команду, що адресує байт, а не слово. При зверненні до старшого байта цієї комірки (з вмістом 2A) процесор виставляє адресу 23421 і використовує команду, що адресує байт. Наступний по порядку 16-розрядний елемент пам'яті з вмістом 487F матиме адресу 23422, тобто знову ж таки парний. Її байти матимуть адреси 23422 і 23423.

Для розрізнення байтових і словних циклів обміну на магістралі в шині управління передбачається спеціальний сигнал байтового обміну. Для роботи з байтами в систему команд процесора вводяться спеціальні команди або передбачаються методи байтової адресації.

Регістри процесора

Як вже згадувалося, внутрішні регістри процесора є надоперативною пам'яттю невеликого розміру, яка призначена для тимчасового зберігання службової інформації або даних. Кількість регістрів в різних процесорах може бути від 6-8 до декількох десятків. Регістри можуть бути універсальними і спеціалізованими. Спеціалізовані регістри, які присутні в більшості процесорів, - це регістр-лічильник команд, регістр стану (PSW), регістр покажчика стека. Решта регістрів процесора може бути як універсальними, так і спеціалізованими. Наприклад, в 16-розрядному процесорі T-11 фірми DEC було 8 регістрів загального призначення (Рон) і один регістр стану. Всі регістри мали по 16 розрядів. З регістрів загального призначення один відводився під лічильник команд, інший - під покажчик стека. Вся решта регістрів загального призначення повністю взаємозамінна, тобто мають універсальне призначення, можуть берегти як дані, так і адреси (покажчики), індекси і т.д. Максимально допустимий об'єм пам'яті для даного процесора складав 64 Кбайт (адреса пам'яті 16-розрядна).

В 16-розрядному процесорі MC68000 фірми Motorola було 19 регістрів: 16-розрядний регістр стану, 32-розрядний регістр лічильника команд, 9 регістрів адреси (32-розрядних) і 8 регістрів даних (32-розрядних). Два регістри адреси відведено під покажчики стека. Максимально допустимий об'єм пам'яті, що адресується, - 16 Мбайт (зовнішня шина адреси 24-розрядна). Всі 8 регістрів даних взаємозамінні. 7 регістрів адреси - теж взаємозамінні. В 16-розрядному процесорі Intel 8086, який став базовим в лінії процесорів, що використовуються в персональних комп'ютерах, реалізований принципово інший підхід. Кожний регістр цього процесора має своє особливе призначення, і замінювати один одного регістри можуть тільки частково або ж не можуть взагалі. Зупинимося на особливостях цього процесора докладніше.

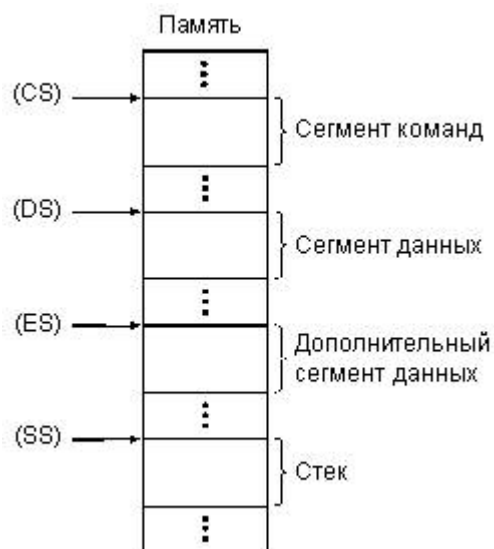
Процесор 8086 має 14 регістрів розрядністю по 16 біт. З них чотири регістри (AX, BX, CX,

DX) - це регістри даних, кожний з яких крім зберігання операндів і результатів операцій має ще і своє специфічне призначення:

- регістр AX - множення, розподіл, обмін з пристроями вводу/виводу (команди введення і висновку);
- регістр BX - базовий регістр в обчисленнях адреси;
- регістр CX - лічильник циклів;
- регістр DX - визначення адреси вводу/виводу.

Для регістрів даних існує можливість роздільного використання обох байтів (наприклад, для регістра AX вони мають позначення AL - молодший байт і AH - старший байт). Наступні чотири внутрішні регістри процесора - це сегментні регістри, кожний з яких визначає положення одного з робочих сегментів (мал. 3.10):

- регістр CS (Code Segment) відповідає сегменту команд, виконуваних в даний момент;
- регістр DS (Data Segment) відповідає сегменту даних, з якими працює процесор;
- регістр ES (Extra Segment) відповідає додатковому сегменту даних;
- регістр SS (Stack Segment) відповідає сегменту стека.

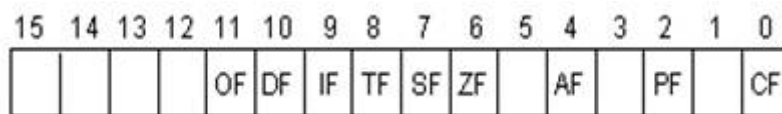


Мал. 3.10. Сегменти команд, даних і стека в пам'яті.

У принципі, всі ці сегменти можуть і перекриватися для оптимального використання простору пам'яті. Наприклад, якщо програма займає тільки частину сегменту, то сегмент даних може починатися відразу після завершення роботи програми (з точністю 16 байт), а не після закінчення всього сегменту програми. Наступні п'ять регістрів процесора (SP - Stack Pointer, BP - Base Pointer, SI - Source Index, DI - Destination Index, IP - Instruction Pointer) служать покажчиками (тобто визначають зсув в межах сегменту). Наприклад, лічильник команд процесора утворюється парою регістрів CS і IP, а покажчик стека - парою регістрів SP і SS. Регістри SI, DI використовуються в рядкових операціях, тобто при послідовній обробці декількох елементів пам'яті однією командою. Останній регістр FLAGS - це регістр стану процесора (PSW). З його 16 розрядів використовуються тільки дев'ять (мал. 3.11):

CF	(Carry Flag)	- прапор перенесення при арифметичних операціях,
PF	(Parity Flag)	- прапор парності результату,
AF	(Auxiliary Flag)	- прапор додаткового перенесення,
ZF	(Zero Flag)	- прапор нульового результату,

SF (Sign Flag) - прапор знака (співпадає із старшим бітом результату),
 TF (Trap Flag) - прапор покрокового режиму (використовується при відладці),
 IF #@: - прапор дозволу апаратних переривань,
 DF #@; - прапор напряму при рядкових операціях, #@



Мал. 3.11. Регістр стану процесора 8086.

Біти регістра стану встановлюються або очищаються залежно від результату виконання попередньої команди і використовуються деякими командами процесора. Біти регістра стану можуть також встановлюватися і очищатися спеціальними командами процесора (про систему команд процесора буде розказано в наступному розділі).

В багатьох процесорах виділяється спеціальний регістр, званий акумулятором (тобто накопичувачем). При цьому, як правило, тільки цей регістр-акумулятор може брати участь у всіх операціях, тільки через нього може проводитися взаємодія з пристроями вводу/виводу. Іноді в нього ж поміщається результат будь-якої виконаної команди (в цьому випадку говорять навіть про «акумуляторну» архітектуру процесора). Наприклад, в процесорі 8086 регістр даних AX можна вважати своєрідним акумулятором, оскільки саме він обов'язково бере участь в командах множення і ділення, а також тільки через нього можна пересилати дані в пристрій вводу/виводу і з пристроєм вводу/виводу. Виділення спеціального регістра-акумулятора спрощує структуру процесора і прискорює пересилки кодів усередині процесора, але в деяких випадках уповільнює роботу системи в цілому, оскільки весь потік інформації повинен пройти через один регістр-акумулятор. У разі, коли декілька регістрів процесора повністю взаємозамінні, таких проблем не виникає.

Система команд процесора

В загальному випадку система команд процесора включає наступні чотири основні групи команд:

- команди пересилки даних;
- арифметичні команди;
- логічні команди;
- команди переходів.

Команди пересилки даних не вимагають виконання ніяких операцій над операндами. Операнди просто пересилаються (точніше, копіюються) з джерела (Source) в приймач (Destination). Джерелом і приймачем можуть бути внутрішні регістри процесора, елемента пам'яті або пристрою вводу/виводу. АЛП в даному випадку не використовується. Арифметичні команди виконують операції додавання, віднімання, множення, ділення, збільшення на одиницю (інкрементування), зменшення на одиницю (декрементування) і т.д. Цим командам потрібні один або два вхідні операнди. Формують команди один вихідний операнд.

Логічні команди проводять над операндами логічні операції, наприклад, логічне І, логічне АБО, виключаюче АБО, очищення, інверсію, різноманітні зсуви (вправо, вліво, арифметичний зсув, циклічний зсув). Цим командам, як і арифметичним, потрібні один або два вхідні операнди, і формують вони один вихідний операнд. Нарешті, команди переходів призначені для зміни звичайного порядку послідовного виконання команд. З їх допомогою організуються переходи на підпрограми і повернення з них, всілякі цикли, розгалуження програм, пропуски фрагментів програм і т.д. Команди

переходів завжди міняють вміст лічильника команд. Переходи можуть бути умовними і безумовними. Саме ці команди дозволяють будувати складні алгоритми обробки інформації. Відповідно до результату кожної виконаної команди встановлюються або очищаються біти регістра стану процесора (PSW). Але треба пам'ятати, що не всі команди змінюють прапори PSW. Це визначається особливостями кожного конкретного процесора. У різних процесорів системи команд істотно розрізняються, але в основі своїй вони дуже схожі. Кількість команд у процесорів також різна. Наприклад, у згаданого вже процесора MC68000 всього 61 команда, а у процесора 8086 - 133 команди. У сучасних могутніх процесорів кількість команд досягає декількох сотень. В той же час існують процесори з скороченим набором команд (так звані RISC-процесори), в яких за рахунок максимального скорочення кількості команд досягається збільшення ефективності і швидкості їх виконання. Розглянемо тепер особливості чотирьох виділених груп команд процесора більш детально.

Команди пересилки даних

Команди пересилки даних займають дуже важливе місце в системі команд будь-якого процесора. Вони виконують наступні найважливіші функції:

- завантаження (запис) вмісту у внутрішні регістри процесора;
- збереження в пам'яті вмісту внутрішніх регістрів процесора;
- копіювання вмісту з однієї області пам'яті в іншу;
- запис в пристрої вводу/виводу і читання з пристроїв вводу/виводу.

В деяких процесорах (наприклад, T-11) всі ці функції виконуються однією єдиною командою MOV (для байтових пересилок - MOV_B) але з різними методами адресації операндів. В інших процесорах крім команди MOV є ще декілька команд для виконання перерахованих функцій. Наприклад, для завантаження регістрів можуть використовуватися команди завантаження, причому для різних регістрів - різні команди (їх позначення звичайно будуються з використанням слова LOAD - завантаження). Часто виділяються спеціальні команди для збереження в стеку і для витягання із стека (PUSH - зберегти в стеку, POP - витягнути із стека). Ці команди виконують пересилку з автоінкрементної і з автодекрементної адресації (навіть якщо ці режими адресації не передбачені в процесорі в явному вигляді).

Іноді в систему команд вводиться спеціальна команда MOVS для рядкової (або ланцюгової) пересилки даних (наприклад, в процесорі 8086). Ця команда пересилає не одне слово або байт, а задану кількість слів або байт (MOVSB), тобто ініціює не один цикл обміну по магістралі, а кілька. При цьому адреса пам'яті, з якою відбувається взаємодія, збільшується на 1 або на 2 після кожного звернення або ж зменшується на 1 або на 2 після кожного звернення. Тобто в неявному вигляді застосовується автоінкремента або автодекремента адресація.

В деяких процесорах (наприклад, в процесорі 8086) спеціально виділяються функції обміну з пристроями вводу/виводу. Команда IN використовується для введення (читання) інформації з пристрою вводу/виводу, а команда OUT використовується для виводу (запису) в пристрій вводу/виводу. Обмін інформацією в цьому випадку проводиться між регістром-акумулятором і пристроєм вводу/виводу. В більш сучасних процесорах цього ж сімейства (починаючи з процесора 80286) додані команди рядкового (ланцюгового) введення (команда INS) і рядкового виводу (команда OUTS). Ці команди дозволяють пересилати цілий масив (рядок) даних з пам'яті в пристрій вводу/виводу або з пристрою вводу/виводу в пам'ять (INS). Адреса пам'яті після кожного обігу збільшується або зменшується #@:. Також до команд пересилки даних відносяться команди обміну інформацією (їх позначення будується на основі слова Exchange). Може бути передбачений обмін інформацією між внутрішніми регістрами, між двома половинами одного регістра (SWAP) або між регістром і елементом пам'яті.

Арифметичні команди

Арифметичні команди розглядають коди операндів як числові двійкові або двійково-десяткові коди. Ці команди можуть бути розділені на п'ять основних груп:

- команди операцій з фіксованою комою (додавання, віднімання, множення, ділення);
- команди операцій з плаваючою комою (додавання, віднімання, множення, ділення);
- команди очищення;
- команди інкремента і декремента;
- команда порівняння.

Команди операцій з фіксованою комою працюють з кодами в регістрах процесора або в пам'яті як із звичайними двійковими кодами. Команда додавання (ADD) обчислює суму двох кодів. Команда віднімання (SUB) обчислює різницю двох кодів. Команда множення (MUL) обчислює добуток двох кодів (розрядність результату удвічі більша розрядності множників). Команда ділення (DIV) обчислює частку від ділення одного коду на інший. Причому всі ці команди можуть працювати як з числами із знаком, так і з числами без знака. Команди операцій з плаваючою комою (крапкою) використовують формат представлення чисел з порядком і мантисою (звичайно ці числа займають два послідовні елементи пам'яті). В сучасних процесорах набір команд з плаваючою комою не обмежується тільки чотирма арифметичними діями, а містить безліч інших складніших команд, наприклад, обчислення тригонометричних функцій, логарифмічних функцій, а також складних функцій, необхідних при обробці звуку і зображення. Команди очищення (CLR) призначені для запису нульового коду в регістр або елемент пам'яті. Ці команди можуть бути замінені командами пересилки нульового коду, але спеціальні команди очищення звичайно виконуються швидше, ніж команди пересилки. Команди очищення іноді відносять до групи логічних команд, але суть їх від цього не міняється.

Команди інкремента (збільшення на одиницю, INC) і декремента (зменшення на одиницю, DEC) також дуже зручні. Їх можна у принципі замінити командами підсумовування з одиницею або віднімання одиниці, але інкремент і декремент виконуються швидше, ніж підсумовування і віднімання. Ці команди вимагають одного вхідного операнда, який одночасно є і вихідним операндом. Нарешті, команда порівняння (позначається CMP) призначена для порівняння двох вхідних операндів. По суті, вона обчислює різницю цих двох операндів, але вихідного операнда не формує, а всього лише змінює біти в регістрі стану процесора (PSW) по результату цього віднімання. Наступна за командою порівняння команда (звичайно це команда переходу) аналізуватиме біти в регістрі стану процесора і виконуватиме дії залежно від їх значень (про команди переходу йдеться в розділі 3.4.4). В деяких процесорах передбачені команди ланцюгового порівняння двох послідовностей операндів, що знаходяться в пам'яті (наприклад, в процесорі 8086 і сумісних з ним).

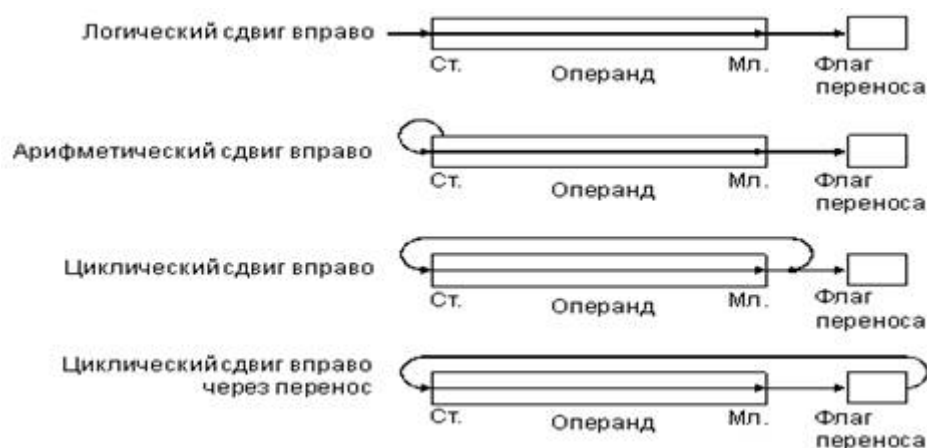
Логічні команди

Логічні команди виконують над операндами логічні (побітові) операції, тобто вони розглядають коди операндів не як єдине число, а як набір окремих бітів. Цим вони відрізняються від арифметичних команд. Логічні команди виконують наступні основні операції:

- логічне І, логічне АБО, складання по модулю 2 (виключаюче АБО);
- логічні, арифметичні і циклічні зсуви;
- перевірка бітів і операндів;

- установка і очищення бітів (прапорів) регістра стану процесора (PSW).

Команди логічних операцій дозволяють побітно обчислювати основні логічні функції від двох вхідних операндів. Крім того, операція І (AND) використовується для примусового очищення заданих бітів (як один з операндів при цьому використовується код маски, в якому розряди, що вимагають очищення, встановлені в нуль). Операція АБО (OR) застосовується для примусової установки заданих бітів (як один з операндів при цьому використовується код маски, в якому розряди, що вимагають установки в одиницю, рівні одиниці). Операція «виключаюче АБО» (XOR) використовується для інверсії заданих бітів (як один з операндів при цьому застосовується код маски, в якому біти, що підлягають інверсії, встановлені в одиницю). Команди вимагають двох вхідних операндів і формують один вихідний операнд. Команди зсувів дозволяють побітно зсовувати код операнда управо (убік молодших розрядів) або вліво (у бік старших розрядів). Тип зсуву (логічний, арифметичний або циклічний) визначає, яке буде нове значення старшого біта (при зсуві управо) або молодшого біта (при зсуві вліво), а також визначає, чи буде десь збережено попереднє значення старшого біта (при зсуві вліво) або молодшого біта (при зсуві управо). Наприклад, при логічному зсуві управо в старшому розряді коду операнда встановлюється нуль, а молодший розряд записується як прапор перенесення в регістр стану процесора. А при арифметичному зсуві управо значення старшого розряду зберігається колишнім (нулем або одиницею), молодший розряд також записується як прапор перенесення. Циклічні зсуви дозволяють зсовувати біти коду операнда по колу (за годинниковою стрілкою при зсуві управо або проти годинникової стрілки при зсуві вліво). При цьому в кільце зсуву може входити або не входити прапор перенесення. В біт прапора перенесення (якщо він використовується) записується значення старшого біта при циклічному зсуві вліво і молодшого біта при циклічному зсуві управо. Відповідно, значення біта прапора перенесення переписуватиметься в молодший розряд при циклічному зсуві вліво і в старший розряд при циклічному зсуві управо. Для прикладу на мал. 3.12 показані дії, виконувані командами зсувів управо. Команди перевірки бітів і операндів призначені для установки або очищення бітів регістра стану процесора залежно від значення вибраних бітів або всього операнда в цілому. Вихідного операнда команди не формують. Команда перевірки операнда (TST) перевіряє весь код операнда в цілому на рівність нулю і на знак (на значення старшого біта), вона вимагає тільки одного вхідного операнда. Команда перевірки біта (BIT) перевіряє тільки окремі біти, для вибору яких як другий операнд використовується код маски. В коді маски бітам основного операнда, що перевіряються, винні відповідати одиничні розряди.



Мал. 3.12. Команди зсувів управо.

Нарешті, команди установки і очищення бітів регістра стану процесора (тобто прапорів) дозволяють встановити або очистити будь-який прапор, що дуже зручно. Кожному прапору звичайно відповідають дві команди, одна з яких встановлює його в одиницю, а інша скидає в

нуль. Наприклад, прапору перенесення C (від Carry) відповідатимуть команди CLC (очищення) і SEC або STC (установка).

Команди переходів

Команди переходів призначені для організації всіляких циклів, розгалужень, викликів підпрограм і т.д., тобто вони порушують послідовний хід виконання програми. Ці команди записують в реєстр-лічильник команд нове значення і тим самим викликають перехід процесора не до наступної по порядку команди, а до будь-якої іншої команди в пам'яті програм. Деякі команди переходів передбачають надалі повернення назад, в точку, з якої був зроблений перехід, інші не передбачають цього. Якщо повернення передбачено, то поточні параметри процесора не зберігаються. Команди переходів без повернення діляться на дві групи:

- команди безумовних переходів;
- команди умовних переходів.

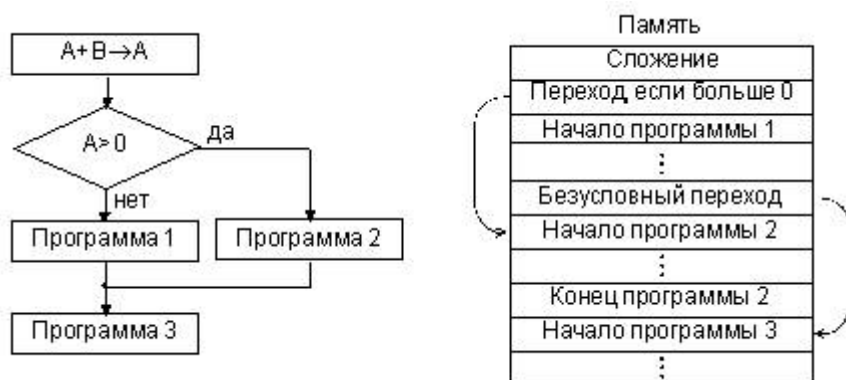
В позначеннях цих команд використовуються слова Branch (галуження) і Jump (стрибок). Команди безумовних переходів викликають перехід на нову адресу незалежно ні від чого. Вони можуть викликати перехід на вказану величину зсуву (вперед або назад) або ж на вказану адресу пам'яті. Величина зсуву або нове значення адреси указуються як вхідний операнд.

Команди умовних переходів викликають перехід не завжди, а тільки при виконанні заданих умов. Як такі умови звичайно виступають значення прапорів в реєстрі стану процесора (PSW). Тобто умовою переходу є результат попередньої операції, що міняє значення прапорів. Всього таких умов переходу може бути від 4 до 16. Декілька прикладів команд умовних переходів:

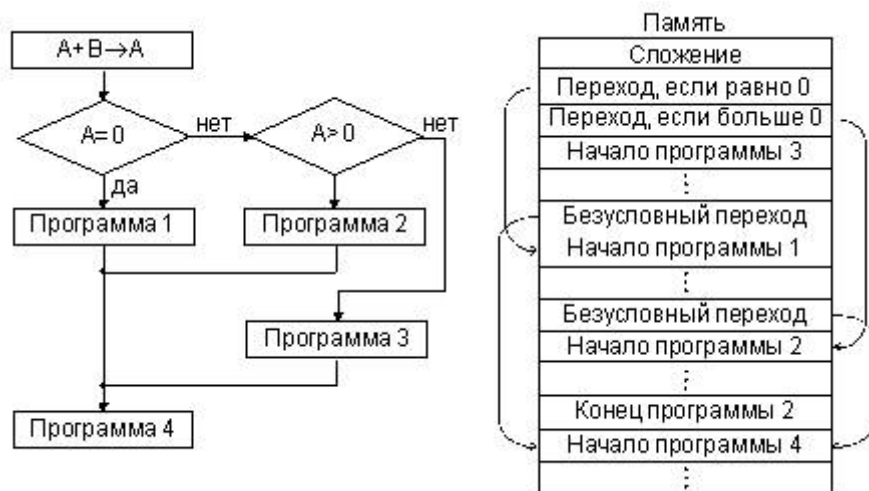
- перехід, якщо рівно нулю;
- перехід, якщо не рівно нулю;
- перехід, якщо є переповнювання;
- перехід, якщо немає переповнювання;
- перехід, якщо більше нуля;
- перехід, якщо менше або рівно нулю.

Якщо умова переходу виконується, то проводиться завантаження в реєстр-лічильник команд нового значення. Якщо ж умова переходу не виконується, лічильник команд просто нарощується, і процесор вибирає і виконує наступну по порядку команду. Спеціально для перевірки умов переходу застосовується команда порівняння (CMP), передуюча команді умовного переходу (або навіть декільком командам умовних переходів). Але прапори можуть встановлюватися і будь-якою іншою командою, наприклад командою пересилки даних, будь-якою арифметичною або логічною командою. Відзначимо, що самі команди переходів прапори не міняють, що якраз і дозволяє ставити декілька команд переходів одну за іншою. Сумісне використання декількох команд умовних і безумовних переходів дозволяє процесору виконувати розгалужені алгоритми будь-якої складності. Для прикладу на мал. 3.13 показано розгалуження програми на дві вітки з подальшим з'єднанням, а на мал. 3.14 - розгалуження на три вітки з подальшим з'єднанням. Команди переходів з подальшим поверненням в точку, з якої був проведений перехід, застосовуються для виконання підпрограм, тобто допоміжних програм. Ці команди називаються також командами виклику підпрограм (поширена назва - CALL). Використовування підпрограм дозволяє спростити структуру основної програми, зробити її

більш логічною, гнучкою, легкою для написання і відладки. В той же час треба враховувати, що широке використання підпрограм, як правило, збільшує час виконання програми.



Мал. 3.13. Реалізація розгалуження на дві вітки.



Мал. 3.14. Реалізація розгалуження на три вітки.

Всі команди переходів з поверненням припускають безумовний перехід (вони не перевіряють ніяких прапорів). При цьому вони вимагають одного вхідного операнда, який може вказуватися як абсолютне значення нової адреси, так і зсув, що складається з поточним значенням адреси. Поточне значення лічильника команд (поточна адреса) зберігається перед виконанням переходу в стеку.

Для зворотного повернення в точку виклику підпрограми (точку переходу) використовується спеціальна команда повернення (RET або RTS). Ця команда витягує із стека значення адреси команди переходу і записує його в регістр-лічильник команд. Особливе місце серед команд переходу з поверненням займають команди переривань (поширена назва - INT). Ці команди як вхідний операнд вимагають номер переривання (адреса вектора). Обслуговування таких переходів здійснюється точно так, як і апаратних переривань. Тобто для виконання даного переходу процесор звертається до таблиці векторів переривань і одержує з неї по номеру переривання адресу пам'яті, в яку йому необхідно перейти. Адреса виклику переривання і вміст регістра стану процесора (PSW) зберігаються в стеку. Збереження PSW - важлива відмінність команд переривання від команд переходів з поверненням.

Команди переривань у багатьох випадках виявляються зручніше, ніж звичайні команди

переходів з поверненням. Сформувати таблицю векторів переривань можна один раз, а потім вже звертатися до неї в міру необхідності. Номер переривання відповідає номеру підпрограми, тобто номеру функції, виконуваною підпрограмою. Тому команди переривання набагато частіше включаються в системи команд процесорів, ніж звичайні команди переходів

з

поверненням.

Для повернення з підпрограми, викликаною командою переривання, використовується команда повернення з переривання (IRET або RTI). Ця команда витягує із стека збережене там значення лічильника команд і регістра стану процесора (PSW). Відзначимо, що у деяких процесорів передбачені також команди умовних переривань, наприклад, команда переривання при переповненні. Звичайно, в даному розділі ми розглянули тільки основні команди, найбільш часто що зустрічаються в процесорах. У конкретних процесорів можуть бути і багато інших команди, що не відносяться до перерахованих груп команд. Але вивчати їх треба вже після того, як вибраний тип процесора, відповідний для задачі, вирішуваною даною мікропроцесорною системою.

Загальні поняття.

Як вже наголошувалося, персональний комп'ютер є найрозвиненішим видом мікропроцесорних систем. На основі персональних комп'ютерів можна будувати найскладніші контрольно-вимірювальні, управляючі, обчислювальні і інформаційні системи. Апаратні і програмні засоби, що є в персональному комп'ютері, роблять його універсальним інструментом для самих різних задач. У разі обчислювальних і інформаційних систем персональний комп'ютер не потребує підключення нестандартної апаратури, все зводиться до підбору або написання необхідного програмного забезпечення. У разі ж контрольно-вимірювальних і управляючих систем персональний комп'ютер оснащується набором інструментів для сполучення із зовнішніми пристроями і відповідними програмними засобами. У багатьох випадках будувати систему на основі персонального комп'ютера виявляється набагато простіше, швидше і навіть дешевше, ніж проектувати її з нуля на базі якогось мікропроцесора, мікропроцесорного комплекту або мікроконтролера. Звичайно, в більшості випадків система на основі персонального комп'ютера виявляється сильно надмірною, це плата за універсальність. Але в той же час один і той же комп'ютер може вирішувати найрізноманітніші задачі. Наприклад, в системі управління технологічними процесами або науковими установками він може математично моделювати процеси, що відбуваються, видавати в реальному часі управляючі сигнали, приймати в реальному часі у відповідь сигнали, накопичувати інформацію, обробляти її, обмінюватися інформацією з іншими комп'ютерами і т.д. Розвинений інтерфейс користувача (відеомонітор, повнорозмірна клавіатура, миша) роблять роботу з персональним комп'ютером комфортною і ефективною. А вартість персональних комп'ютерів унаслідок великого об'єму випуску постійно знижується. Тому їх використання не тільки зручне, але і економічно вигідно. Але щоб грамотно і повноцінно використовувати персональний комп'ютер у складі будь-яких систем, треба мати уявлення про його архітектуру, про основні принципи побудови, про пристрої, що входять в його склад, нарешті, про зовнішні інтерфейси.

Архітектура персонального комп'ютера

Персональний комп'ютер типу IBM PC має досить традиційну архітектуру мікропроцесорної системи і містить всі звичайні функціональні вузли: процесор, постійну і оперативну пам'ять, пристрої вводу/виводу, системну шину, джерело живлення.

Основні особливості архітектури персональних комп'ютерів зводяться до принципів компоновки апаратури, а також до вибраного набору системних апаратних засобів.



Мал. 7.1. Архітектура персонального комп'ютера типа IBM PC.

Функції основних вузлів комп'ютера наступні:

- **Центральний процесор** - це мікропроцесор зі всіма необхідними допоміжними мікросхемами, включаючи зовнішню кеш-пам'ять і контролер системної шини. (Про кеш-пам'ять докладніше розказано в наступних розділах). В більшості випадків саме центральний процесор здійснює обмін по системній шині.
- **Оперативна пам'ять** може займати майже весь простір пам'яті процесора, що адресується. Проте частіше всього її об'єм набагато менше. В сучасних персональних комп'ютерах стандартний об'єм системної пам'яті складає, як правило, від 64 до 512 Мбайт. Оперативна пам'ять комп'ютера виконується на мікросхемах динамічної пам'яті і тому вимагає регенерації.
- **Постійна пам'ять** (ROM BIOS - Base Input/Output System) має невеликий об'єм (до 64 Кбайт), містить програму початкового запуску, опис конфігурації системи, а також драйвери (програми нижнього рівня) для взаємодії з системними пристроями.
- **Контролер переривань** перетворює апаратні переривання системної магістралі в апаратні переривання процесора і задає адреси векторів переривання. Всі режими функціонування контролера переривань задаються програмно процесором перед початком роботи.
- **Контролер прямого доступу до пам'яті** приймає запит на ПДП з системної магістралі, передає його процесору, а після надання процесором магістралі проводить пересилку даних між пам'яттю і пристроєм вводу/виводу. Всі режими функціонування контролера ПДП задаються програмно процесором перед початком роботи. Використовування вбудованих в комп'ютер контролерів переривань і ПДП дозволяє істотно спростити апаратуру вживаної плати розширення.
- **Контролер регенерації** здійснює періодичне оновлення інформації в динамічній оперативній пам'яті шляхом проведення по шині спеціальних циклів регенерації. На час циклів регенерації він стає господарем (задатчиком) шини.
- **Перестановщик байтів** даних допомагає проводити обмін даними між 16-розрядним і 8-розрядним пристроями, пересилати цілі слова або окремі байти.
- **Годинник реального часу і таймер-лічильник** - це пристрої для внутрішнього контролю часу і дати, а також для програмної витримки тимчасових інтервалів, програмного задання частоти і т.д.
- **Системні пристрої вводу/виводу** - це ті пристрої, які необхідні для роботи комп'ютера і взаємодії із стандартними зовнішніми пристроями по паралельному і

послідовному інтерфейсам. Вони можуть бути виконані на материнській платі, а можуть розташовуватися на платі розширення.

- **Плата розширення** встановлюється в слоти (роз'єми) системної магістралі і може містити оперативну пам'ять і пристрої вводу/виводу. Вони можуть обмінюватися даними з іншими пристроями на шині в режимі програмного обміну, в режимі переривань і в режимі ПДП. Передбачена також можливість захоплення шини, тобто повного відключення від шини всіх системних пристроїв на деякий час.

Важлива особливість подібної архітектури - її відкритість, тобто можливість включення в комп'ютер додаткових пристроїв, причому як системних пристроїв, так і плат розширення. Відкритість припускає також можливість простого вбудовування програм користувача на будь-якому рівні програмного забезпечення комп'ютера. Перший комп'ютер сімейства, що набув широке поширення, IBM PC XT, був виконаний на базі оригінальної системної магістралі PC XT-Bus. Надалі (починаючи з IBM PC AT) вона була дорацьована до магістралі, стандартної, що стала, і одержала назву ISA (Industry Standard Architecture). До недавнього часу ISA залишалася основою комп'ютера. Проте, починаючи з появи процесорів i486 (в 1989 році), вона перестала задовольняти вимогам продуктивності, і її стали дублювати більш швидкими шинами: VLB (VESA Local Bus) і PCI (Peripheral Component Interconnect bus) або замінювати сумісною з ISA магістраллю EISA (Enhanced ISA). Поступово шина PCI витіснила конкурентів і стала фактичним стандартом, а починаючи з 1999 роком в нових комп'ютерах рекомендується повністю відмовлятися від магістралі ISA, залишаючи тільки PCI. Правда, при цьому доводиться відмовлятися від вживання плати розширення, розробленої за довгі роки для підключення до магістралі ISA. Інший напрям вдосконалення архітектури персонального комп'ютера пов'язаний з максимальним прискоренням обміну інформацією з системною пам'яттю. Саме з системної пам'яті комп'ютер читає всі виконувані команди, і в системній же пам'яті він береже дані. Тобто більш всього звертань процесор скоює саме до пам'яті. Прискорення обміну з пам'яттю приводить до істотного прискорення роботи всієї системи в цілому. Але при використуванні для обміну з пам'яттю системної магістралі доводиться враховувати швидкісні обмеження магістралі. Системна магістраль повинна забезпечувати сполучення з великим числом пристроїв, тому вона повинна мати досить велику протяжність; вона вимагає вживання вхідних і вихідних буферів для узгодження з лініями магістралі. Цикли обміну по системній магістралі складні, і прискорювати їх не можна. В результаті істотного прискорення обміну процесора з пам'яттю по магістралі добитися неможливо. Розробниками був запропонований наступний підхід. Системна пам'ять підключається не до системної магістралі, а до спеціальної високошвидкісної шини, що знаходиться «ближче» до процесора, не вимагаючої складних буферів і великих відстаней. У такому разі обмін з пам'яттю йде з максимально можливою для даного процесора швидкістю, і системна магістраль не уповільнює його. Особливо актуальним це стає із зростанням швидкодії процесора (зараз тактові частоти процесорів персональних комп'ютерів досягають 2-4 ГГц). Таким чином, структура персонального комп'ютера з одношинної, застосовувалася тільки в перших комп'ютерах, стає трьохшинною.

Призначення шин наступне:

- до локальної шини підключаються центральний процесор і кеш-пам'ять (швидка буферна пам'ять);
- до шини пам'яті підключається оперативна і постійна пам'ять комп'ютера, а також контролер системної шини;
- до системної шини (магістралі) підключається вся решта пристроїв комп'ютера.



Мал. 7.2. Організація зв'язків у разі трьохшинної структури.

Всі три шини мають адресні лінії, лінії даних і управляючі сигнали. Але склад і призначення ліній цих шин не співпадають між собою, хоча вони і виконують однакові функції. З погляду процесора, системна шина (магістраль) в системі всього одна, по ній він одержує дані і команди і передає дані як в пам'ять, так і в пристрої вводу/виводу. Тимчасові затримки між системою пам'яттю і процесором в даному випадку мінімальні, оскільки локальна шина і шина пам'яті сполучені тільки найпростішими швидкодійними буферами. Ще менше затримки між процесором і кеш-пам'яттю, що підключається безпосередньо до локальної шини процесора і що служить для прискорення обміну процесора з системою пам'яттю.



Мал. 7.3. Приклад багатощинної структури.

Якщо в комп'ютері застосовуються дві системні шини, наприклад, ISA і PCI, то кожна з них має свій власний контролер шини, і працюють вони паралельно, не впливаючи один на одного. Тоді виходить вже чотирьохшинна, а іноді і п'ятишинна структура. Приклад такої структури комп'ютера приведений на мал. 7.3.

В найпоширеніших настільних комп'ютерах класу Desk-top як конструктивна основа використовується системна або материнська платна (motherboard), на якій розташовуються всі основні системні вузли комп'ютера, а також декілька роз'ємів (слотів) системної шини, для підключення дочірньої плати - плати розширення (інтерфейсних модулів, контролерів, адаптерів). Як правило, сучасна системна плата допускає заміну процесора, вибір його тактової частоти, заміну і нарощування оперативної пам'яті, вибір режимів роботи інших

вузлів.

На системній платі зараз звичайно розташовуються також основні засоби зовнішнього інтерфейсу, що служать для приєднання як вбудованих пристроїв (наприклад, дискових носіїв), так і зовнішніх пристроїв комп'ютера (наприклад, клавіатури, миші, принтера, сканера, модему). Для підключення відеомонітора, як правило, використовується спеціальна платна розширення (контролер дисплея), що вставляється в один із слотів. Це дозволяє замінювати її більш могутньою при необхідності установки нового монітора.

Відзначимо, що для одержуючих все більш широке розповсюдження портативних персональних комп'ютерів класу ноутбуків (notebook) застосовуються дещо інші конструктивні рішення. Зокрема, в них відсутні слоти розширення системної шини, а практично всі вузли комп'ютера виконуються на одній платі. Але ми в основному говоритимемо про комп'ютери типу desktop (настільних), оскільки саме вони найбільш пристосовані для побудови складних систем, допускають досить просту модернізацію (upgrade) і настройку на конкретні потреби користувача.

Процесори персональних комп'ютерів

Не дивлячись на те, що перший персональний комп'ютер був випущений фірмою Apple, зараз персональними комп'ютерами називають в основному IBM PC-сумісні комп'ютери. Це зв'язано, перш за все, з тим, що фірма IBM вибрала правильну ринкову політику: вона не приховувала принципів пристрою своїх комп'ютерів і не патентувала основних рішень. В результаті багато виробників стали випускати сумісні комп'ютери, і вони швидко стали фактичним стандартом. Через великий об'єм випуску персональні комп'ютери почали швидко дешевшати. До того ж для IBM-сумісних персональних комп'ютерів стали розробляти безліч програмних засобів, що ще більше сприяло їх розповсюдженню. Тому, не дивлячись на деякі істотні архітектурні недоліки, IBM-сумісні персональні комп'ютери зараз упевнено займають перше місце на ринку. Із самого початку фірма IBM орієнтувалася на процесори Intel. У цих процесорів були дуже могутні конкуренти, наприклад, процесори фірм Motorola або Zilog, що перевершували процесори Intel по багатьох параметрах, але саме завдяки персональним комп'ютерам процесори Intel змогли вийти переможцями в конкурентній боротьбі. До того ж ще деякі фірми (наприклад, AMD, VIA, Cyrix) випускають Intel-сумісні процесори. Тому ми розглянемо основні особливості процесорів фірми Intel. Це дозволить нам також прослідити основні тенденції в розвитку процесорів за останні десятиріччя. Свій перший 16-розрядний процесор i8086 фірма Intel випустила в 1978 році. Він міг адресувати 1 Мбайт пам'яті (тобто мав 20-розрядну шину адреси), продуктивність його при тактовій частоті 5 Мгц складала 0,33 MIPS, але незабаром з'явилися процесори з тактовою частотою 8 і 10 Мгц. Трохи пізніше (через рік) була випущена спрощена версія процесора i8086 - i8088, який відрізнявся тільки 8-розрядною зовнішньою шиною даних. За рахунок цього він був повільніше, ніж i8086, на 20-60% при тій же тактовій частоті, та зате помітно дешевше. Саме на його основі був зібраний дуже популярний персональний комп'ютер IBM PC XT.

16-розрядний процесор i80286, випущений в 1982 році, був використаний в персональних комп'ютерах IBM PC AT. Принципово новим було в ньому те, що він міг адресувати до 16 Мбайт пам'яті і мав крім реального режиму, аналогічного i8086, ще і так званий захищений режим, що дозволяє більш гнучко управляти пам'яттю. Продуктивність цього процесора при тактовій частоті 8 Мгц складала 1,2 MIPS. Важливим кроком стала поява в 1985 році повністю 32-розрядного процесора i80386, здатного адресувати до 4 Гбайт пам'яті (32-розрядна адресна шина). Він мав ще більш розвинену систему управління пам'яттю MMU (Memory Management Unit). Продуктивність його при тактовій частоті 16 Мгц складала 6 MIPS. З появою цього процесора стала бурхливо розвиватися операційна система MS Windows, істотно що змінила процес роботи з

Кожна команда містить 1, 2 або 4 байти коду команди, за якими можуть слідувати 1, 2 або 4 байти операнда.

В процесорі передбачені програмні і апаратні переривання, розділення зовнішньої шини з іншими процесорами або з контролером прямого доступу до пам'яті, а також можливість підключення математичного співпроцесора i8087, істотно збільшуючого продуктивність обчислень.

При старті процесора (по зовнішньому сигналу RESET) він переходить на адресу пам'яті FFFF0 і починає виконання програми, яка розміщується починаючи з цією адресою. Процесор може обробляти 256 типів переривань: зовнішніх (апаратних), програмних і внутрішніх. Вектори переривань є подвійним словом (два слова по 16 розрядів), що визначає сегмент і зсув початкової адреси програми обробки переривань. Для векторів переривань відведена область пам'яті з адресами 00000.003FF. Внутрішні переривання виробляються при особливих ситуаціях:

- переривання 0 відповідає переповнюванню при поділі на нуль;
- переривання 1 виробляється після кожної команди при встановленому прапорі трасування TF в регістрі стану процесора (див. розділ 3.2);
- переривання 4 виробляється по спеціальній команді INTO, якщо встановлений прапор переповнювання в регістрі стану процесора (це умовне переривання по переповнюванню).

Особливе місце займає немасковане переривання NMI (Non-Masked Interrupt), яке виробляється під час приходу зовнішнього сигналу NMI і не залежить від стану прапора дозволу апаратних переривань IF. В комп'ютері воно використовується для контролю парності пам'яті, контролю коректності обміну з пам'яттю і пристроями вводу/виводу, а також для обробки так званих виключень, тобто особливих умов, що виникають в процесі роботи. Немаскованим воно називається саме тому, що його не можна заборонити. Важлива відмінна особливість процесора - розділення операцій обміну з пристроями вводу/виводу і з пам'яттю. Для обміну з пристроями вводу/виводу використовуються як окремі команди введення і виведення, так і спеціальні управляючі сигнали на шині управління. Адреси і дані як при обміні з пам'яттю, так і при обміні з пристроями вводу/виводу передаються по одних і тих же шинах. Але якщо для обміну з пам'яттю використовуються всі 20 розрядів шини адреси (адресується 1 Мбайт - адреси 00000 ... FFFFF), то в циклах обміну з пристроями вводу/виводу - тільки 16 розрядів шини адреси (адресується 64 Кбайта - адреси 00000 ... 0FFFF). Такий підхід має як свої переваги (наприклад, спрощення реалізації прямого доступу до пам'яті), так і недоліки (ускладнення системи команд, збільшення кількості управляючих сигналів). Мікропроцесори i8086/8088 виконані у вигляді інтегральної мікросхеми в 40-вивідному корпусі. Відмінність в призначенні виводів мікросхеми між ними тільки одна: адреса в процесорі 8088 не мультиплексована з даними (передається по окремих лініях), а в процесорі 8086 - мультиплексована.

Процесор працює від одного джерела живлення напругою +5В і вимагає зовнішнього тактуючого сигналу з частотою, визначуваною номером моделі (від 4,77 МГц до 10 МГц). Спеціальний управляючий сигнал MN/MX визначає мінімальний або максимальний режим роботи процесора. В мініальному режимі процесор сам виробляє сигнали управління для зовнішньої шини. Цей режим використовується для побудови найпростіших систем. Для роботи у складі комп'ютера застосовується максимальний режим, при якому сигнали управління зовнішньою шиною виробляються спеціальною мікросхемою контролера шини i8288.

Особливості процесора 80386

32-розрядний процесор i80386 відкрив новий етап в історії мікропроцесорів Intel і персональних комп'ютерів типу IBM PC. Природно, він зберігав повну сумісність з своїми 16-розрядними попередниками, щоб не відмовлятися від розробленого для них програмного забезпечення. Але саме в 80386 подолано жорстке обмеження на довжину безперервного сегменту пам'яті в 64 Кбайт, що було пережитком минулого і слідством не найвдаліших архітектурних рішень 8086.

В захищеному режимі 80386 довжина сегменту може досягати 4 Гбайт, тобто всього об'єму пам'яті, що фізично адресується. Таким чином, пам'ять фактично стала безперервною. Крім того, 80386 забезпечує підтримку віртуальної пам'яті об'ємом до 64 Тбайт (1 Тбайт = 1024 Гбайт). Вбудований блок управління пам'яттю підтримує механізми сегментації і сторінкової трансляції адрес (Paging). Забезпечується чотирьохрівнева система захисту пам'яті і вводу/виводу, а також перемикання задач.

Процесор 80386, як і 80286, може працювати в двох режимах: Реальний режим, який повністю сумісний з 8086. Захищений режим. В цьому режимі можлива адресація до 4 Гбайт фізичної пам'яті (32 розряди), через які при використуванні механізму сторінкової адресації може відображатися до 16 Тбайт віртуальної пам'яті кожної задачі. Перемикання між цими двома режимами в обидві сторони, на відміну від 80286, проводиться достатньо швидко, за допомогою простої послідовності команд, і апаратного скидання процесора не вимагається.

Процесор може оперувати з 8, 16, 32-бітовими операндами, рядками байт, слів і подвійних слів, а також з бітами, бітовими полями і рядками біт. В архітектуру процесора введені засоби відладки і тестування. Розрядність регістрів даних (AX, BX, CX, DX) і адрес (SI, DI, BP, SP) збільшена до 32. При цьому в їх позначенні з'явилася приставка E (Extended - розширений), наприклад, EAX, ESI. Відсутність приставки в імені означає посилання на молодші 16 розрядів відповідного регістра. Регістри даних і адрес з'єднані в групу регістрів загального призначення, які іноді можуть замінювати один одного. Це може розглядатися як відхід від ідеології спеціалізації всіх регістрів.

Регістри селекторів залишені 16-розрядними, але додано два нові регістри FS і GS для завдання додаткових сегментів даних. Також розширений до 32 розрядів регістр-показчик (лічильник) команд EIP. 32-розрядним став і регістр прапорів EFLAGS. Його біти, визначені для 8086 і 80286, залишилися колишніми, але додано 6 нових біт (мал. 7.6). Такий же формат використовується і в процесорах 80486 і Pentium.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	NT	IOPL	OF	DF	IF	TF	SF	ZF	0	AF	0	PF	1	CF	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	ID	VIP	VIF	AC	VM	RF

Мал. 7.6. Регістр ознак EFLAGS процесора 80386.

Регістри сегментів процесора містять 16-бітові показники (в реальному режимі) або селектори (в захищеному режимі) шести сегментів. З кожним з шести сегментних регістрів пов'язані програмно недоступні регістри дескрипторів, як і у випадку 80286. В захищеному режимі в регістри дескрипторів завантажується 32-бітова базова адреса сегменту, 32-бітовий ліміт і атрибуту сегментів.

Крім згаданих регістрів до складу процесора входять ще чотири управляючі регістри (CR0, CR1, CR2, CR3), які бережуть ознаки стану процесора, загальні для всіх задач. В процесорі Pentium до них доданий ще і регістр CR4. Крім того, процесор містить ще системні адресні регістри для посилань на сегменти і таблиці в захищеному режимі, регістри відладки і регістри тестування. Як бачимо, від моделі до моделі кількість регістрів процесора постійно

зростає.

Процесор дозволяє виділяти в пам'яті сегменти і сторінки. Сегменти в реальному режимі мають фіксований розмір, в захищеному - змінний. Сторінки, яких не було в попередніх моделях, є областями логічної пам'яті розміром 4 Кбайт, кожна з яких може відобразитися на будь-яку область фізичної пам'яті. Якщо сегменти використовуються на прикладному рівні, то сторінки застосовуються на системному. Стосовно пам'яті розрізняють три адресні простори: логічний, лінійний і фізичний. Про принципи адресації пам'яті у процесора 80386 вже мовилося в розділі 3.1.2 (див. мал. 3.8). Процесор 80386 може використовувати режими 32-бітової або 16-бітової адресації. Режим 16-бітової адресації відповідає режимам процесорів 8086 і 80286, при цьому як компоненти адреси використовуються молодші 16 біт відповідних регістрів. Відмінність 32-бітової адресації відображена в табл. 7.1. Так само, як і попередники, процесор 80386 обробляє всі види переривання: апаратні (масковані і немасковані) і програмні, які в даному випадку обробляються як різновид виключень, і власне виключення. Виключення підрозділяються на відмови, пастки і аварійні завершення.

Таблиця 7.1. Відмінності режимів адресації 80386.

Компонент адресація	16-бітна адресація	32-бітна
Базовий регістр	BX або BP	Будь-який 32-бітовий загального призначення
Індексний регістр	SI або DI	Будь-який 32-бітовий загального призначення, окрім ESP
Масштаб	Ні (завжди 1)	1, 2, 4 або 8
Зсув	0, 8 або 16 біт	0, 8 або 32 біта

- **Відмова (Fault)** - це виключення, яке виявляється і обслуговується до виконання команди, що викликає помилку.
- **Пастка (Trap)** - це виключення, яке виявляється і обслуговується після виконання команди, що викликає це виключення. До класу пасток відносяться і програмні переривання.
- **Аварійне завершення (Abort)** - це виключення, яке не дозволяє точно встановити команду, що викликала його. Воно використовується для повідомлення про серйозну помилку, такий як апаратна помилка або пошкодження системних таблиць.

Кожному номеру переривання (0.255) або виключення відповідає елемент в таблиці дескрипторів переривань IDT (Interrupt Descriptor Table). В захищеному режимі IDT може мати розмір від 32 до 256 дескрипторів, кожен з яких складається з 8 байт. Відмінності від попереднього процесора 80286 у виконанні операцій вводу/виводу зводяться до додавання можливостей звернення до 32-бітових портів. Важливо відзначити, що рядкові команди процесора 80386 забезпечують блокове вводу/виводу з більшою швидкістю, ніж стандартний контролер прямого доступу до пам'яті. Процесор випускався в 100-вивідному корпусі. Була передбачена можливість підключення зовнішнього співпроцесора 80387. А зараз зупинимося трохи докладніше на захищеному режимі, який використовується на повну потужність саме починаючи з 32-розрядними процесорами, оскільки процесор 80286 мав істотні обмеження і в більшості випадків працював все-таки в реальному режимі.

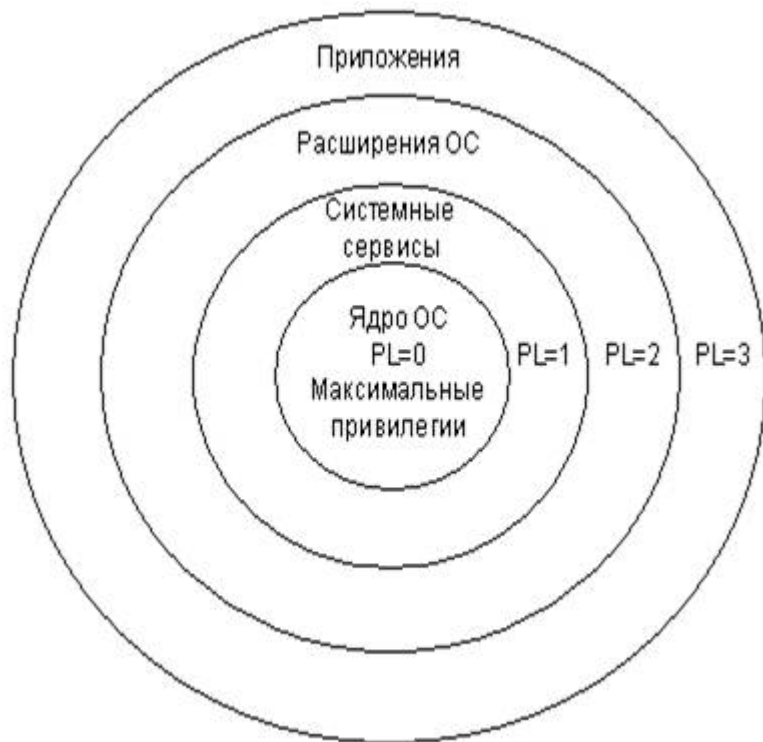
Захищений режим був запропонований для забезпечення незалежності одночасного виконання декількох задач (як системних, так і прикладних). Для цього передбачений захист ресурсів кожної задачі від дій інших задач. Під ресурсами тут розуміється пам'ять з даними, програмами, системними таблицями, а також апаратура, що використовується задачею. Захист ґрунтується на сегментації пам'яті, причому, на відміну від реального режиму, задача не може перевизначати положення своїх сегментів в пам'яті і використовує тільки сегменти, визначені для неї операційною системою. Сегмент визначається дескриптором сегменту, який задає положення сегменту в пам'яті, його розмір (або ліміт), призначення і характеристики захисту.

Захист за допомогою сегментації не дозволяє:

- використовувати сегменти не за призначенням, наприклад, потрактувати область даних як область програми;
- порушувати права доступу (наприклад, намагатися записувати інформацію в сегмент, призначений тільки для читання, або звертатися до сегменту, не маючи достатніх привілеїв);
- адресуватися до елементів, що виходять за межу сегменту;
- змінювати дескриптори сегментів, не маючи достатніх привілеїв.
- Захищений режим передбачає засоби перемикання задач. Стан кожної задачі (тобто стан всіх реєстрів процесора) зберігається в спеціальному сегменті стану задачі, на який указує селектор в реєстрі задачі. При перемиканні задачі достатньо завантажити в реєстр задачі новий селектор, і стан попередньої задачі автоматично збережеться, а в процесор завантажиться стан нової (або раніше перерваної) задачі. Це розвиток ідеї стека.

В захищеному режимі передбачається ієрархічна чотирьохрівнева (рівні 0, 1, 2, 3) система привілеїв, призначена для управління виконанням привілейованих команд і доступом до дескрипторів (мал. 7.7). Рівень 0 відповідає необмеженим правам доступу і відводиться ядру операційної системи. Рівень 3 дає мінімальні права і відводиться прикладним задачам. Рівні привілеїв відносяться до дескрипторів, селекторів і задач. Крім того, в реєстрі прапорів є двобітове поле привілеїв вводу/виводу, що управляє доступом до команд вводу/виводу і прапором переривань.

Механізм віртуальної пам'яті, що використовується в захищеному режимі, дозволяє будь-якій задачі використовувати логічний простір розміром до 64 Тбайт (16К сегментів по 4 Гбайта). Для цього кожний сегмент в своєму дескрипторі має спеціальний біт, вказуючий на присутність даного сегменту в оперативній пам'яті у нинішній момент. невживаний сегмент може бути вивантажений з оперативної пам'яті в зовнішню пам'ять (звичайно - на диск), про що робиться помітка в його дескрипторі. На місце, що звільнилося, із зовнішньої пам'яті може закачуватися інший сегмент (це називається свопінгом або підкачкою). При зверненні задачі до відсутнього в оперативній пам'яті сегменту виробляється спеціальне виключення, яке і виконує свопінг. З погляду виконуваної програми, віртуальна пам'ять нічим не відрізняється від реальної (говорять, що віртуальна пам'ять прозора), не рахуючи затримки на процес перекачування інформації на диск і з диска.



Мал. 7.7. Рівні привілеїв 32-розрядних процесорів.

Реальне використання системи захисту і віртуальної пам'яті покладається на операційну систему, яка в ідеалі повинна забезпечувати працездатність навіть у разі некоректного виконання прикладних задач.

В пам'яті існує три типи таблиць дескрипторів: локальна таблиця дескрипторів LDT, глобальна таблиця дескрипторів GDT і таблиця дескрипторів переривань IDT. Кожній таблиці відповідає свій регістр процесора (відповідно, LDTR, GDTR і IDTR), де зберігаються дескриптори сегментів. Глобальна таблиця містить дескриптори, доступні всім задачам, а локальна може бути для кожної задачі своя. Дескриптори складаються з 8 байтів (як і у 80286). Проте призначення байтів різне. Для прикладу на мал. 7.8 показані формати дескрипторів сегменту програм і даних процесорів 80286 і 80386.

Існують також системні сегменти, призначені для зберігання локальних таблиць дескрипторів і таблиць стану задач. Їх дескриптори (теж 8-байтні) визначають базову адресу, ліміт сегменту, права доступу (читання, читання/запис, тільки виконання або виконання/читання) і присутність сегменту в оперативній пам'яті.



Мал. 7.8. Дескриптори сегментів програми і даних.

Задачі, дескриптори і селектори мають свої рівні привілеїв. Привілеї задач діють на виконання команд і використання дескрипторів. Поточний рівень привілею задач визначається двома молодшими бітами регістра CS. Привілеї дескриптора описуються полем DPL (мал. 7.8). DPL визначає якнайменший рівень привілеїв, з яким можливий доступ до даного дескриптора. Привілеї селектора задаються полем RPL (див. мал. 7.5). Привілеї перевіряються при спробах запису в сегментні регістри, а також при виконанні деяких команд.

Таким чином, починаючи з процесором 80386, з'являються засоби обслуговування багатозадачного режиму. Природно, процесор не може обробляти декілька задач одночасно, виконуючи по декілька команд відразу. Він тільки періодично перемикається між задачами. Але з погляду користувача виходить, що комп'ютер паралельно працює з декількома задачами.

Особливості процесорів Pentium

Процесори Pentium відносяться до п'ятого покоління процесорів або до третього покоління 32-розрядних процесорів. За своїми основними архітектурними принципами вони сумісні з процесорами 386 і 486. Але є істотні відмінності, що дозволяють говорити про нове покоління:

- Суперскалярна архітектура процесора, тобто процесор має два п'ятиступінчатих паралельно працюючих конвейера обробки інформації, завдяки чому він здатний одночасно виконувати дві команди за один такт. Необхідно відзначити, що переваги такої архітектури виявляються тільки у разі спеціальної компіляції програмного забезпечення, що дозволяє здійснювати паралельну обробку.
- Зовнішня 64-розрядна шина даних для підвищення продуктивності. Це вимагає відповідної організації пам'яті. Через таку особливість процесор іноді неправильно називають 64-розрядним (хоча всередині він все-таки залишився 32-розрядним). Зовнішня шина адреси процесора - 32-розрядна.
- Вживання технології динамічного прогнозу розгалужень (переходів).
- Роздільний кеш для команд і даних об'ємом 8 Кбайт кожний. Довжина рядка кеша - 32 байти. Обидва кеші працюють в режимі зворотного запису.
- Підвищена в 2-10 разів в порівнянні з процесором 486 продуктивність вбудованого математичного співпроцесора. В ньому застосована восьмиступінчата конвейеризація

і спеціальні блоки складання, множення і ділення, що дозволяє виконувати операції з плаваючою комою за один такт процесора.

- Скорочений час (число тактів) виконання команд.
- Передбачена можливість побудови двопроцесорних систем.
- Введені засоби управління енергоспоживанням і тестування.

Прогноз розгалужень дозволяє продовжувати вибірку і декодування потоку команд після вибірки команди розгалуження (переходу), не чекаючи перевірки умови переходу. В колишніх моделях в даному випадку доводилося припинити конвейєр. Динамічний прогноз ґрунтується на аналізі попередньої програми і накопиченні статистики поведінки. Виходячи з цього аналізу передбачається найвірогідніша умова кожного переходу, що зустрічається в програмі.

На додаток до базової архітектури 32-розрядних процесорів Pentium має набір регістрів MSR (Model Specific Registers). В нього входить група тестових регістрів (TR1 - TR12), засоби стеження за продуктивністю, регістри-фіксатори адреси і даних циклу, що викликав спрацьовування контролю машинної помилки. Назва цієї групи регістрів указує на їх унікальність для кожної моделі процесорів Pentium. Засоби для стеження за продуктивністю (моніторингу продуктивності) включають таймер реального часу і лічильники подій. Таймер є 64-розрядним лічильником, що інкрементується з кожним тактом процесора. Два лічильники подій мають розрядність по 40 біт і програмується на підрахунок подій різних класів, пов'язаних з шинними операціями, виконанням команд, пов'язаних з роботою кеша, контролем точок зупинки і т.д. Порівнюючи стани таймера і лічильників подій, можна зробити висновок про продуктивність процесора. Тестові регістри дозволяють управляти більшістю функціональних вузлів процесора, забезпечуючи можливість докладного тестування їх працездатності. Спеціальні біти регістра TR12 дозволяють відключити нові архітектурні властивості (прогноз і трасування галузень, паралельне виконання інструкцій), а також роботу первинного кеша. Засоби для побудови двопроцесорних систем дозволяють на одній локальній шині встановлювати два процесори з об'єднанням майже всіх однойменних виводів. Це дає можливість використовувати симетричну мультипроцесорну обробку (SMP - Symmetric Multi-Processing) або будувати функціонально надмірні системи (FRC - Functional Redundancy Checking).

В режимі SMP кожний процесор виконує свою задачу, доручену йому операційною системою, що підтримується такими системами, як Novell NetWare, Windows NT, Unix. Обидва процесори розділяють загальні ресурси комп'ютера, включаючи пам'ять і пристрої вводу/виводу. В кожний момент шиною управляє один процесор, за певними правилами вони міняються ролями. В ідеальному випадку продуктивність системи збільшується удвічі (без урахування звернень до шини і часу на перемикання процесорів). В конфігурації FRC два процесори виступають як один логічний процесор. Основний процесор (Master) працює в звичайному однопроцесорному режимі. Перевірочний процесор (Checker) виконує всі ті ж операції усередині себе, не управляючи зовнішньою шиною, і порівнює сигнали основного процесора з тими, які генерує сам. У разі неспівпадання формується сигнал помилки, оброблюваний як переривання. Тобто в даному випадку збільшується надійність системи в цілому (в ідеалі - удвічі). Розвитком Pentium стало додавання технології MMX, розрахованої на мультимедійне, графічне і комунікаційне вживання. Основна ідея MMX полягає в одночасній обробці декількох елементів даних за одну команду (SIMD, Single Instruction - Multiple Data). Розширення MMX використовує нові типи упакованих 64-бітових даних:

- упаковані байти - вісім байт;
- упаковані слова - чотири слова;
- упаковані подвійні слова (два подвійні слова);
- збільшене учетверо слово (одне слово).

Ці типи даних можуть оброблятися у восьми додаткових 64-розрядних регістрах MMX0 - MMX7. В систему команд для підтримки MMX введено 57 додаткових команд для одночасної обробки декількох одиниць даних (команди пересилки, арифметичні, логічні команди і команди перетворення форматів даних). Команди MMX доступні з будь-якого режиму процесора.

Крім того, в процесорах Pentium MMX збільшений об'єм кеша даних і програм (до 16 Кбайт кожний), збільшено число ступенів конвейєрів і введено ще декілька удосконалень, що підвищують продуктивність звичайних (не мультимедійних) операцій. Від процесора Pentium Pro прийнято відлічувати шосте покоління процесорів. Pentium Pro в порівнянні з Pentium має наступні удосконалень:

- Динамічне виконання команд припускає, що команди, не залежні від результатів попередніх операцій, можуть виконуватися в зміненому порядку, проте послідовність обміну із зовнішніми пристроями (пам'яттю і пристроями вводу/виводу) відповідатиме програмі. Тобто процесор сам вибирає зручний йому порядок виконання команд. Це дозволяє підвищити продуктивність процесора без збільшення тактової частоти.
- Архітектура подвійної незалежної шини підвищує сумарну пропускну здатність. Одна шина (системна) служить для обміну з основною пам'яттю і пристроями вводу/виводу, а інша (локальна) призначена тільки для обміну з вторинним кешем (див. мал. 7.3).
- В процесор введений кеш другого рівня об'ємом 256-512 Кбайт.
- Можлива побудова багатопроцесорних систем (до чотирьох мікропроцесорів).

Процесор Pentium II поєднує в собі архітектуру Pentium Pro з технологією MMX. Розмір первинних кешів даних і команд складає 16 Кбайт, вторинного кеша - до 512 Кбайт. Кеш другого рівня дещо повільніший, ніж кеш першого рівня, зате він має більший об'єм. Така дворівнева організація дозволяє досягати компромісу між швидкістю кеш-пам'яті і її об'ємом.

Шина адреси має 36 розрядів (тобто максимально допустимий об'єм пам'яті - 64 Гбайта). Процесори Pentium III і Pentium 4 відрізняються значно збільшеною граничною тактовою частотою (до 3 ГГц у Pentium 4, а в перспективі і до 5 ГГц), збільшеним об'ємом кеша (від 512 Кбайт до декількох мегабайт) і подальшим вдосконаленням архітектури Pentium. Розмір внутрішнього конвейєра у Pentium 4 доведений до 20 ступенів. Вже з'явилися і повністю 64-розрядні процесори. Правда, повне використання можливостей 64-розрядної архітектури зажадає істотну зміну програмного забезпечення. Треба відзначити, що у складі персональних комп'ютерів практично ніколи не використовуються всі можливості процесорів сімейства Pentium. Наприклад, багатопроцесорні системи зустрічаються достатньо рідко, а об'єм системної пам'яті лише іноді перевищує 512 Мбайт.

Для портативних комп'ютерів були запропоновані спрощені версії процесорів Pentium III і Pentium 4, що продаються під маркою Celeron. Вони відрізняються зменшеною тактовою частотою і скороченим об'ємом кеша другого рівня. Їх тактова частота зараз доходить до 3 ГГц. Треба враховувати, що розсіювана потужність процесора Celeron залишається практично такою ж, як у процесорів Pentium III і Pentium 4.

Пам'ять персонального комп'ютера

Як і в будь-якій іншій мікропроцесорній системі, пам'ять персонального комп'ютера складається з двох частин: оперативної пам'яті і постійної пам'яті. Обидві частини розташовані в адресному просторі пам'яті, до обох комп'ютер може звертатися однаково чиним. Обидві пам'яті допускають звернення до окремих байтів, 16-розрядним словам (має парні адреси), до 32-розрядних подвійних слів (має адреси, кратні чотирьом) і до 64-розрядних збільшених учетверо слів (має адреси, кратні восьми). Відмінність тільки в тому,

що оперативну пам'ять використовують для тимчасового зберігання програм і даних, а в постійній пам'яті зберігаються програми початкового запуску, початкового самотестування комп'ютера, а також набір програм вводу/виводу нижнього рівня, тобто те, що не повинне втрачатися при виключенні живлення комп'ютера. Об'єм оперативної пам'яті набагато більший, ніж постійної пам'яті.

Оперативна пам'ять

Оперативна пам'ять займає значну частину адресного простору комп'ютера. Її встановлений об'єм і швидкодія роблять величезний вплив на швидкість персонального комп'ютера в цілому (деколи навіть більше, ніж швидкість процесора). Надійність її роботи багато в чому визначає надійність всього комп'ютера. Тому всіма розробниками їй надається велика увага. Всі персональні комп'ютери використовують оперативну пам'ять динамічного типу (DRAM - Dynamic Random Access Memory), основною перевагою якої перед статичною оперативною пам'яттю (SRAM - Static RAM) є низька ціна. Це зв'язано з тим, що якщо елемент статичної пам'яті (тригер) вимагає 4-6 транзисторів, то елемент динамічної пам'яті - це інтегральний конденсатор, для обслуговування якого потрібні 1-2 транзистори. Звідси ж слідує два основні недоліки динамічної пам'яті: вона вимагає регенерації (тобто постійного відновлення заряду на конденсаторі, що розряджається) і має у декілька разів меншу швидкість у порівнянні із статичною пам'яттю. До того ж під час регенерації динамічна пам'ять неприступна для обміну, що також знижує швидкість комп'ютера. Відзначимо, що зараз звичайно застосовують вбудовану регенерацію, що не вимагає зовнішнього обслуговування, але знову-таки займаючи час. Перехід на повністю статичну пам'ять дуже сильно підвищив би вартість комп'ютера в цілому (до того ж необхідний об'єм оперативної пам'яті комп'ютера все збільшується), тому статичну пам'ять застосовують тільки в «найвужчих» місцях, там, де без високої швидкості не обійтися, наприклад, для кеш-пам'яті. Що стосується динамічної пам'яті, той її розвиток йде по шляху зниження часу доступу завдяки структурним і технологічним удосконаленням. Наприклад, друге покоління динамічної пам'яті EDO RAM (Extended Data Output RAM) мало швидкість приблизно на 20-25% вище, ніж у звичайної пам'яті. Це досягається за рахунок того, що наступне звернення до пам'яті можливе ще до завершення попереднього обігу. Третє покоління динамічної пам'яті - SDRAM (Synchronous DRAM) - ще на стільки ж швидше (робоча частота в даний час досягає 133 МГц). Ще більш швидкісна пам'ять - DDR SDRAM (частота до 400 МГц) і пам'ять RDRAM (частота до 1 ГГц і навіть вище). Для спрощення установки оперативної пам'яті в комп'ютер її виконують у вигляді невеликих модулів - друкарської плати з ножовим (друкарським) роз'ємом, на яку встановлені мікросхеми пам'яті (SIMM - Single In-Line Memory Module). Ці модулі встановлюються в спеціальні SIMM-роз'єми на материнській платі, що дозволяє користувачу легко міняти об'єм пам'яті комп'ютера, враховуючи при цьому необхідний рівень швидкості, складність вирішуваних задач і свої фінансові можливості. Широко застосовуються 72-контактні SIMM-модулі різного об'єму. Відзначимо, що не рекомендується одночасно використовувати декілька модулів з різною швидкістю: в деяких комп'ютерах це приводить до збоїв. В сучасних комп'ютерах на базі процесорів Pentium застосовуються також модулі DIMM (Dual In-Line Memory Module - модуль пам'яті з двосторонніми друкарськими висновками), що мають 64 біти даних. Модулі пам'яті іноді підтримують перевірку вмісту пам'яті на парність. Для цього до 8 біт даних кожної адреси пам'яті додається дев'ятий біт парності. Він записується при кожному записі інформації у відповідний байт пам'яті і перевіряється при читанні інформації з відповідного байта пам'яті. Якщо виявляється невідповідність біта парності байту інформації, то виробляється сигнал, що викликає немасковане переривання NMI. Весь адресний простір пам'яті комп'ютера розділяється на декілька областей, що зв'язане, в першу чергу, з необхідністю забезпечення сумісності з першими комп'ютерами сімейства. В комп'ютері IBM PC XT на процесорі i8088 процесор міг адресувати 1 Мбайт пам'яті (20

адресних розрядів). Але всі програмні і апаратні засоби будувалися виходячи з припущення, що доступний адресний простір - тільки молодші 640 Кбайт (тоді це здавалося цілком достатнім). Дана область пам'яті одержала назву стандартної пам'яті (Conventional memory). Саме в межах ці 640 Кбайт (адреси 0..9FFFF) працює операційна система MS DOS і всі її прикладні програми.

Перші 1024 байти (адреси 0..3FF) зберігають таблицю векторів переривання (Interrupt Vectors) об'ємом 256 подвійних слів, формовану на етапі початкового завантаження. Проте якщо процесор працює в захищеному режимі, таблиця векторів може розташовуватися в будь-якому іншому місці пам'яті.

Адреси 400..4FF відводяться під область змінних BIOS (BIOS Data Area). Докладніше про BIOS буде розказано в наступному розділі.

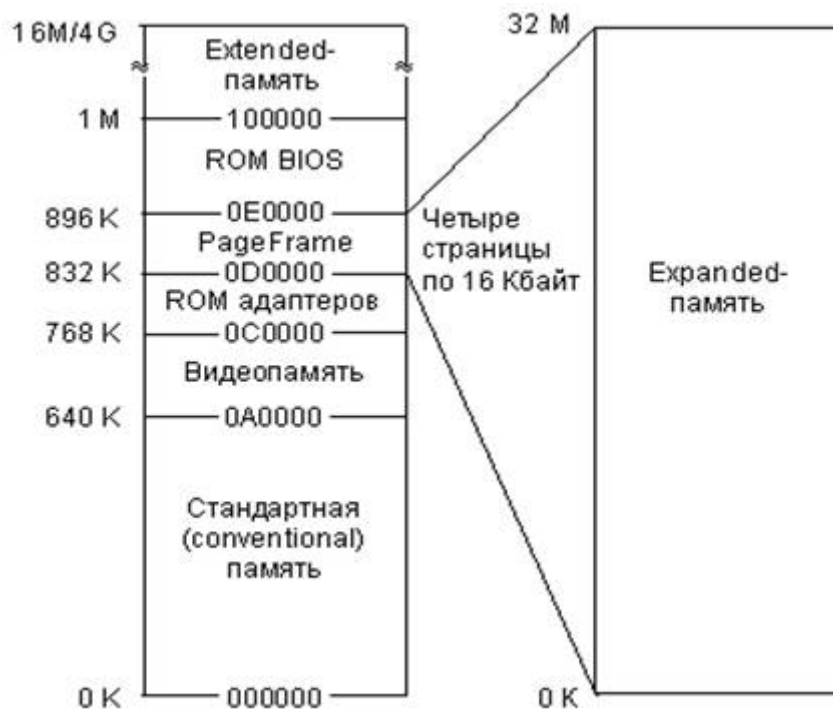
Адреси 500..9FFFF включають область операційної системи DOS (DOS Area) і пам'ять користувача (User RAM).

Решта від 1 Мбайта пам'яті, а це 384 Кбайта (адреси A0000..FFFFFF), зарезервовані під інші системні потреби, називаються UMA (Upper Memory Area) - область верхньої пам'яті або UMB (Upper Memory Blocks) - блоки верхньої пам'яті або High DOS Memory. Простір відеопам'яті (адреси A0000..BFFFF) містить області для зберігання текстової і графічної інформації відеоадаптера.

Простір пам'яті з адресами E0000..FFFFFF відведено під системну постійну пам'ять комп'ютера ROM BIOS.

В цій же області виділено вікно розміром в 64 Кбайта (page frame) з адресами D0000..DFFFF, через яке програми могли діставати доступ до додаткової (що відображається) пам'яті (Expanded memory) об'ємом до 32 Мбайт, залишаючись в межах того ж 1 Мбайта пам'яті, що адресується. Це досягається шляхом почергового відображення чотирьох сторінок по 16 Кбайт з додаткової пам'яті у виділене вікно. При цьому положення сторінок в додатковій пам'яті можна змінювати програмним шляхом. Зрозуміло, що працювати з додатковою пам'яттю менш зручно, ніж з основною, оскільки в кожний момент комп'ютер «бачить» тільки вікно в 64 Кбайт. Тому зараз вона застосовується досить рідко. В даний час область пам'яті з адресами C0000..DFFFF частіше використовується для оперативної і постійної пам'яті, що входить до складу різних адаптерів і плати розширення комп'ютера.

В результаті логічна організація адресного простору в межах 1 Мбайт вийшла досить складною (мал. 7.10). І таку ж організацію повинні підтримувати всі персональні комп'ютери сімейства IBM PC для забезпечення сумісності з попередніми моделями.



Мал. 7.10. Розподіл адресного простору пам'яті.

При подальшому розширенні простору пам'яті в подальших моделях комп'ютерів, вся пам'ять об'ємом понад 1 Мбайт одержала назву розширеної пам'яті (Extended memory). Для доступу до неї мікропроцесор повинен переходити з реального режиму в захищений і назад. Загальний об'єм пам'яті персонального комп'ютера (верхня межа розширеної пам'яті) може доходити до 16 Мбайт (24 розряди адреси) або до 4 Гбайт (32 розряди адреси). Особливої уваги заслуговує так звана тінюва пам'ять (Shadow RAM), частина оперативної пам'яті, в яку при запуску комп'ютера переписується вміст постійної пам'яті, і яка змінюється під час роботи комп'ютера. Необхідність даної процедури викликана тим, що навіть порівняно повільна динамічна оперативна пам'ять виявляється все-таки швидшою, ніж постійна пам'ять. Постійна пам'ять часто помітно стримує швидкодію комп'ютера. Тому було запропоновано виділяти частину оперативної пам'яті для виконання обов'язків як системної постійної пам'яті ROM BIOS, так і постійної пам'яті, що входить до складу додаткових адаптерів, які підключаються до комп'ютера. Переписування інформації звичайно передбачено в програмі початкового пуску. У зв'язку з особливостями роботи динамічної пам'яті для скорочення часу доступу до неї застосовуються спеціальні режими роботи оперативної пам'яті: режим розшарування (інтерлівінг) і сторінковий режим. Використовування режиму інтерлівінгу припускає не зовсім звичайне розбиття пам'яті на банки (частини). Якщо при звичайному розбитті (послідовної адресації) адреси наступного банку починаються після закінчення адрес попереднього, то при інтерлівінгу адреси банків чергують. Тобто, наприклад, після першої адреси першого банку слідує перша адреса другого банку, потім друга адреса першого банку і друга адреса другого банку і т.д. Виходить, що в одному банку парні слова, а в іншому - непарні. Таких чергуючих банків може бути не тільки два, а чотири, вісім, шістнадцять. Об'єми банків при цьому повинні бути однаковими. В результаті такого підходу з'являється можливість починати звернення до наступного слова ще до закінчення процесу доступу до попереднього. Сторінковий режим припускає посторінкову роботу мікросхем пам'яті, коли вибір сторінки проводиться один раз на всю сторінку, а вибір комірки усередині сторінки може відбуватися набагато швидше. Для підтримки обох режимів застосовуються спеціальні технологічні рішення.

Тут же відзначимо, що при заміні пам'яті комп'ютера або при установці додаткових банків пам'яті треба строго слідувати рекомендаціям виготівників системної плати, оскільки порядок заповнення банків може не бути далеко очевидний.

Постійна пам'ять

Системна постійна пам'ять (ПЗП) займає порівняно невеликий об'єм (звичайно 64 Кбайта). Проте її значення для комп'ютера дуже велике. Сама її назва ROM BIOS (ROM Basic Input/Output System) - базова система вводу/виводу - говорить про те, що в ній знаходиться функціонально повний набір програм нижнього рівня для управління пристроями вводу/виводу. Тому навіть до завантаження в оперативну пам'ять виконуваних програм з диска комп'ютер має нагоду обслуговувати клавіатуру, дисплей, подавати звукові сигнали, спілкуватися з дисками і т.д. Важливо відзначити, що більшість сучасних відеоадаптерів, а також контролерів накопичувачів мають власну систему BIOS, яка доповнює або навіть замінює системну BIOS під час основної роботи. Виклик програм BIOS здійснюється через програмні або апаратні переривання, для чого BIOS формує відповідну таблицю векторів переривань. Але цим функції постійної пам'яті не обмежуються. У принципі, під пам'ять ROM BIOS відведено 128 Кбайт адресного простору пам'яті. В перших комп'ютерах (IBM PC XT) вона займала всього 8 Кбайт, зараз звичайно займає 64 Кбайт. Якщо ж потрібно використовувати системну постійну пам'ять більшого об'єму, то вона по черзі відображається на вікно системної пам'яті розміром 64 Кбайт (адреси F0000...FFFFFF). Це пов'язано з прагненням заощадити простір верхньої пам'яті для інших цілей.

При старті комп'ютера після включення живлення, натиснення кнопки на передній панелі RESET або після програмного перезавантаження починає виконуватися програма початкового запуску, що також зберігається в постійній пам'яті (початкова адреса FFFF0). Ця програма включає:

- програму самотестування комп'ютера (POST - Power On Self Test);
- програму початкового завантаження операційної системи з відповідного дискового накопичувача комп'ютера;
- програму завдання поточних параметрів комп'ютера (Setup).

Окрім програми початкового запуску ROM BIOS також обслуговує апаратні переривання від системних пристроїв (таймера, клавіатури, дисків), а також відпрацьовує базові програмні звернення до системних пристроїв. Відзначимо, що останнім часом ROM BIOS виконується не на мікросхемах власне постійній пам'яті, а на мікросхемах, що допускають багатократний перезапис інформації користувачем EPROM (Erasable Programmable ROM) з електричним перезаписом - типу флеш (flash), що дозволяє більш гнучко налаштувати комп'ютер. Користувач може легко модернізувати BIOS свого комп'ютера за допомогою програми перезапису флеш-пам'яті. При використуванні ж мікросхем ПЗП з ультрафіолетовим стиранням для цього були потрібні спеціальний програматор і стираючий пристрій (ультрафіолетова лампа). Програма самотестування POST дозволяє проводити найпростішу діагностику основних вузлів комп'ютера, включаючи визначення повного об'єму встановленої оперативної пам'яті. Інформація про її результати виводиться трьома способами:

- повідомленнями на дисплеї (найбільш наочно і зрозуміло користувачу);
- звуковими сигналами (що дуже корисне при непрацездатному дисплеї);
- видачею кодів в певний порт вводу/виводу, звичайно це порт з адресою 080 (на цьому засновані всі діагностичні прилади).

Повідомлення на дисплеї можуть прямо називати знайдену несправність або несправний блок або ж видавати спеціальний код помилки, по якому згодом в прикладеній документації

можна знайти причину несправності комп'ютера. Звукові сигнали не відрізняються особливою різноманітністю, але, проте, дозволяють знайти і ідентифікувати основні помилки. Використовування спеціальних діагностичних приладів особливо ефективно: за деякими оцінками, з їх допомогою можна знайти до 95% несправностей. Проте їх застосовують тільки спеціальні співробітники сервісних служб. Починаючи з комп'ютерів на процесорі 80286 (IBM PC AT) постійна пам'ять ROM BIOS обов'язково доповнюється невеликою енергозалежною оперативною пам'яттю CMOS RAM, яка виконана на мікросхемах із зниженим енергоспоживанням з технологією КМОП (CMOS) і при виключенні живлення комп'ютера живиться від батареї або акумулятора (ця пам'ять, як правило, входить до складу інших мікросхем). В CMOS-пам'яті зберігається інформація про поточні дані годинника (дата і час), про значення часу для будильника, про конфігурацію комп'ютера: пріоритети завантаження з різних накопичувачів, кількості пам'яті, типах накопичувачів, режимах енергоспоживання, про тип дисплея, про установки клавіатури і т.д. CMOS RAM відрізняється від постійної пам'яті тим, що записана в неї інформація легко міняється програмним шляхом. Задавати всі параметри комп'ютера, що зберігаються в CMOS RAM, дозволяє програма BIOS Setup, викликати яку можна шляхом натиснення призначених клавіш під час процедури початкового завантаження комп'ютера (інформація про це завжди виводиться на екран). В сучасних комп'ютерах дана програма пропонує досить зручний і наочний інтерфейс користувача із звичними меню. Опис роботи з BIOS Setup будь-якого комп'ютера обов'язково поставляється разом з ним. Іноді за допомогою цієї програми вдається значно підвищити швидкість комп'ютера завдяки вибору оптимальних (або навіть граничних) для даної конфігурації параметрів: частоти системної шини, кількості тактів затримки при обміні з системною пам'яттю і кеш-пам'яттю. Програма Setup дозволяє також дозволити або заборонити використання тіншової (Shadow) пам'яті як для системного BIOS, так і для BIOS адаптерів, що використовуються (по окремих сегментах пам'яті). При використуванні тіншової пам'яті у вибрану область оперативної пам'яті копіюється вміст BIOS ROM, ця область визначається як доступна тільки для читання, і проводиться переадресація пам'яті. Тобто при будь-якому звертанні за адресами ROM читання даних проводитиметься з відповідних адрес Shadow RAM, а постійна пам'ять вже не використовується. Такий підхід може істотно (іноді в 4 - 5 разів) прискорити вибірку команд для програм обміну з відеоадаптером і з дисковими накопичувачами. В програмі Setup завжди передбачена можливість установки параметрів комп'ютера за умовчанням (Default Setting). Це особливо зручно у разі розряду або пошкодження батареї або акумулятора.

В нових комп'ютерах, що підтримують режим економії споживаної електроенергії, можна також задавати перехід комп'ютера в режими Doze (сплячий), Standby (очікування або резервний) і Suspend (припинення роботи) за відсутності звернень до вузлів комп'ютера протягом заданого часу. Режими перераховані в порядку зниження споживання електроенергії. Комп'ютери (а також їх системна плата), де застосовуються такі режими, називаються іноді «зеленими».

Системні пристрої

Крім мікропроцесора і системної пам'яті на системній (материнській) платі розташовуються і інші важливі модулі, що забезпечують працездатність комп'ютера: контролери переривань і прямого доступу, тактовий генератор, системний таймер, буферні мікросхеми, контролер шини і т.д. В перших комп'ютерах сімейства всі ці функції виконували окремі спеціалізовані мікросхеми порівняно низкою ступеня інтеграції. В сучасних комп'ютерах застосовуються надвеликі інтегральні схеми, які, проте, забезпечують повну програмну і апаратну сумісність з попередніми моделями. Ці мікросхеми називаються набором мікросхем або chipset

(чипсет). Переваги такого підходу - chipset займає менше місця на платі, менша споживана потужність, значно велика надійність. Комп'ютери, виконані на системній платі з chipset відомих фірм, мають кращу репутацію. В більшість наборів мікросхем входить так званий периферійний контролер, що включає два контролери переривань, два контролери прямого доступу до пам'яті, таймер, годинник реального часу, а також CMOS-пам'ять.

Тактовий генератор

Системний тактовий генератор генерує сигнали синхронізації для роботи мікропроцесора, всіх контролерів і системної шини. Для забезпечення високої стабільності тактових частот і їх незалежності від температури застосовуються кварцові резонатори, тобто кристали кварцу, що мають високостабільну частоту власних коливань. Як правило, до складу системної плати входить декілька кварцових резонаторів, кожний з яких забезпечує свій тактовий сигнал. Тактову частоту процесора можна вибирати шляхом установки перемичок на системній платі. Це дозволяє користувачу модернізувати комп'ютер шляхом заміни процесора на інший, розрахований на більш високу тактову частоту. Іноді вдається примусити процесор працювати на більш високій тактовій частоті, ніж та, на яку він розрахований, але тут потрібно дотримуватися обережності, оскільки підвищення частоти веде не тільки до збільшення споживаної потужності і перегріву мікросхеми, але і до помилок і збоїв в роботі, причому які виявляються тільки в окремих режимах, наприклад, в багатозадачному.

В комп'ютерах на базі процесорів 486 і Pentium застосовується розподіл опорної тактової частоти для синхронізації системної шини і внутрішнє множення частоти в процесорах. Наприклад, в процесорі 486DX2-66 використовується множення на два тактової частоти 33,3 МГц, а в процесорах сімейства Pentium застосовується множення на 2,5 (при опорній частоті 60 МГц частота процесора 150 МГц) або на 3 (при опорній частоті 66,6 МГц частота процесора 200 МГц).

В загальному випадку в комп'ютері існують наступні тактові частоти:

- Host Bus Clock (CLK2IN) - це опорна частота (зовнішня частота шини процесора). Саме з неї можуть виходити інші частоти і саме вона задається перемичками (джамперами);
- CPU Clock (Core Speed) - це внутрішня частота процесора, на якій працює його обчислювальне ядро. Може співпадати з Host Bus Clock або виходити з неї множенням на 1,5, 2, 2,5, 3, 4. Множення повинне бути передбачено в конструкції процесора.
- ISA Bus Clock (ATCLK, BBUSCLK) - це тактова частота системної шини ISA (сигнал SYSCLOCK). За стандартом вона повинна бути близька до 8 МГц, але в BIOS Setup є можливість вибрати її через коефіцієнт розподілу частоти Host Bus Clock. Іноді комп'ютер залишається працездатним і при частоті шини ISA близько 20 МГц, але звичайно плата розширення ISA розробляється з розрахунку на 8 МГц, і при великих частотах вони перестають працювати. Не слід розраховувати, що комп'ютер стане удвічі швидше при подвоєнні цієї частоти. Для каналів прямого доступу до пам'яті на системній платі використовується ще один тактовий сигнал SCLK, частота якого, як правило, складає половину від ISA Bus Clock.
- PCI Bus Clock - це тактова частота системної шини PCI, яка за стандартом повинна бути 25 - 33,3 МГц. Її звичайно одержують розподілом частоти Host Bus Clock на потрібний коефіцієнт. В комп'ютерах передбачається можливість її збільшення до 75 або навіть 83 МГц, але з міркувань надійності роботи рекомендується дотримуватися стандартних значень.
- VLB Bus Clock - це частота локальної шини VLB, визначується аналогічно PCI Bus Clock.

Контролер переривань

Контролери радіальних переривань в перших комп'ютерах виконувалися на мікросхемах i8259, кожна з яких мала 8 входів запиту переривань. В IBM PC AT застосовувалося дві такі мікросхеми, внаслідок чого кількість входів запитів переривань збільшилася до 15. Режими роботи контролерів переривань визначаються процесором шляхом запису відповідних управляючих кодів за адресами в просторі пристроїв вводу/виводу. Про цикли обміну по перериваннях вже мовилося в розділі 2.2.2. Під час приходу запиту переривання на один з входів IRQ і задоволенні цього запиту контролер переривань виробляє вихідний сигнал запиту переривання, що поступає на процесор. У відповідь на це процесор опитує контролер про те, переривання з якою адресою вектора необхідно обслужити. Всього переривань може бути 256 (від 00 до FF). Номер переривання, одержаний від контролера, процесор м на 4 і одержує, таким чином, адресу пам'яті, де зберігається адреса початку програми обробки переривань (вектор переривання). Привласнення кожному з каналів IRQ свого номера процесор здійснює на етапі ініціалізації контролера. Контролер переривань може виконувати наступний набір операцій:

- маскуванню запитів на переривання, тобто тимчасова заборона реакції на них;
- установка пріоритетів запитів по різних входах, тобто дозвіл конфліктів при одночасному приході декількох запитів на переривання;
- робота як основний контролер (Master) або підлеглий (Slave).

Для маскуванню переривань використовується внутрішній регістр контролера, програмно доступний процесору як по запису, так і по читанню. Замаскований може бути кожний запит (по кожній з ліній IRQ), шляхом установки відповідного біта маскуванню в записуваному в контролер управляючому байті. Схема пріоритетів переривань може бути задана процесором програмним шляхом. В базовому варіанті всі пріоритети фіксовані (тобто IRQ0 має вищий пріоритет, а IRQ7 - низький). Але у принципі вищий рівень пріоритету задається для будь-якого з входів запитів, можна також встановити циклічне перемикання пріоритетів (останній обслужений запит одержує низький пріоритет), забезпечуючи тим самим всім запитам рівні пріоритети. Якщо під час обробки переривання з меншим рівнем пріоритету приходить більш пріоритетний запит, то процесор переходить на програму обробки більш пріоритетного запиту, а після її виконання повертається до програми обробки менш пріоритетного запиту. Відзначимо, що немасковане переривання NMI має пріоритет вище за будь-яке інше апаратне переривання. Завершивши виконання програми обробки переривання, процесору необхідно за допомогою спеціальної команди повідомити про це контролер переривань, щоб дозволити йому подальшу роботу, зокрема, знов обробляти той же самий запит. Контролер 8259 передбачає можливість вибору способу розпізнавання запиту на переривання - по фронту сигналу IRQ і по рівню цього сигналу. В архітектурі комп'ютера типа PC використовується запит по фронту. При каскадуванні контролерів основному контролеру треба вказати, до якого з його входів підключений підлеглий контролер, а підлеглому контролеру необхідно вказати, до якого входу основного контролера підключений його вихідний сигнал запиту.

Всі операції початкової настройки контролерів переривань виконує BIOS, і користувачу потрібно вдаватися до програмування цих контролерів тільки при необхідності зміни режимів обслуговування переривань або при написанні власної програми обробки апаратних переривань.

Контролер прямого доступу до пам'яті

Контролер прямого доступу до пам'яті має складнішу структуру в порівнянні з контролером переривань, що пов'язане з його складнішими функціями. На час ПДП контролер ПДП стає

задатчиком (активним пристроєм) системної шини, виставляючи всі основні сигнали шини самостійно. Проте всі режими роботи контролера ПДП, всі його параметри встановлюються процесором на етапі ініціалізації контролера. Зокрема, процесор визначає тип вирішуваної задачі, задає початкову адресу передаваного або приймається масиву даних, а також розмір цього масиву.

В персональному комп'ютері застосовується два чотирьохканальні контролери ПДП типу i8237, що забезпечують 7 каналів ПДП (один канал задіяний для каскадування контролерів за тим же принципом, що і контролери переривань). Крім власне контролерів ПДП в підсистему ПДП входять також програмно доступні регістр старшого байта адреси і регістри сторінок ПДП.

Про цикли обміну в режимі ПДП вже мовилося в розділі 2.2.3. Одержавши сигнал запиту ПДП по одній з ліній DRQ, контролер формує запит процесору на захоплення шини і, одержавши дозвіл від процесора, повідомляє про надання ПДП пристрою, що запитав прямий доступ, по відповідній лінії DACK. Після цього проводиться цикл ПДП, що пересилає дані з пристрою вводу/виводу в пам'ять або навпаки. При цьому сам контролер ПДП формує тільки 16 молодших розрядів адреси пам'яті, а вісім старших розрядів містяться в регістрі сторінок. Свій власний регістр сторінок відповідає кожному з каналів ПДП. Передача даних у разі ПДП можлива по одному з наступних режимів:

- Режим одиночної (поциклової) передачі. В цьому випадку за час надання ПДП виконується тільки один цикл передачі даних, і для наступної передачі треба знову ж таки запитати ПДП. Проте адреса пам'яті, з якою здійснюється обмін, автоматично міняється після кожного циклу. Цей режим дозволяє процесору вставляти свої цикли обміну після кожного циклу ПДП.
- Режим передачі блоку (режим блокової передачі). В цьому режимі за один раз передається цілий блок даних певної довжини (до 64 Кбайт). Режим забезпечує більш високу швидкість передачі в порівнянні з режимом одиночної передачі, але може на довгий час захопити системну шину, не допускаючи до її управління процесор.
- Режим передачі на вимогу. Цей режим дозволяє продовжувати ПДП до тих пір, поки пристрій, що запитав ПДП, не вичерпає весь об'єм даних.
- Каскадний режим дозволяє об'єднувати контролери для збільшення кількості каналів ПДП.

У принципі, можливий режим передачі в режимі ПДП з пам'яті в пам'ять, але в комп'ютері він не використовується.

Як і у разі контролера переривань, можливі дві схеми пріоритетів каналів ПДП, вибрані програмно, - фіксований і циклічний. Початкова схема - фіксовані пріоритети, причому нульовий канал має максимальний пріоритет, а сьомий - мінімальний. Під час обслуговування будь-якого запиту ПДП решта запитів не може втрутитися, але після завершення обслуговування даного запиту обслуговуватиметься запит з найбільшим пріоритетом. Як і у разі контролера переривань, можливе програмне маскування кожного каналу ПДП.

Перед початком роботи кожному з каналів контролера процесор повинен вказати режим, в якому він працюватиме, базову адресу пам'яті, з якої почнеться обмін, кількість передаваних байтів або слів, напрям запису, в пам'ять або читання з пам'яті (від старших адрес до молодших або навпаки). Можливий режим автоініціалізації, при якому після закінчення пересилки всього масиву даних контролер автоматично відновлює всі параметри, і для цього не потрібне втручання процесора. Передбачена також можливість програмного запиту ПДП, обслуговуваного так само, як і апаратний запит.

Системний таймер і годинник реального часу

Як системний таймер комп'ютера використовується мікросхема трьохканального 16-розрядного таймера i8254 або її аналог. Таймер призначений для отримання програмно керованих тимчасових затримок і генерації сигналів заданої частоти. Таймер дозволяє підвищити ефективність програмування процесів управління і синхронізації зовнішніх пристроїв, особливо в реальному часі. Таймер містить три незалежні канали (0, 1 і 2), кожний з яких може бути, у принципі, запрограмований на роботу в одному з шести режимів рахунку:

- переривання термінального рахунку;
- запрограмований генератор одиночного імпульсу;
- генератор імпульсів заданої частоти;
- генератор прямокутних імпульсів типу «меандр»;
- програмно формований строб;
- апаратно формований строб.

На кожному канал можуть подаватися вхідні синхросигнали. З кожного каналу можна одержувати сигнал з частотою, рівній вхідній частоті, діленій на довільне 16-розрядне число. В комп'ютері на всі входи поступають синхросигнали частотою 1,19 МГц. Всі канали таймера в комп'ютері мають спеціальне призначення, тому особливої свободи у виборі режиму роботи або перепризначенні функцій у користувача немає. Вихід каналу 0 пов'язаний з сигналом запиту переривання IRQ0 і забезпечує переривання для лічильника реального часу (використовується режим роботи 3). Користувачу не рекомендується перепрограмувати цей канал. При старті комп'ютера канал програмується так, щоб видавати імпульси приблизно 18,2 рази в секунду. По цьому перериванню програмно збільшується стан лічильника реального часу. Користувач може читати стан даного лічильника із спеціально виділеного елемента пам'яті і застосовувати його для затримок в своїх програмах. Вихід каналу 1 генерує сигнал запиту регенерації динамічної пам'яті (режим роботи 2). Використовування цього каналу не за призначенням може привести до втрати вмісту оперативної пам'яті. Вихід каналу 2 генерує тональний сигнал для вбудованого динаміка комп'ютера (режим роботи 3). Проте дозвіл цього тонального сигналу проводиться установкою виділених розрядів (0 і 1) програмно доступного паралельного порту контролера периферійних пристроїв. Один розряд (0) дозволяє роботу каналу, інший розряд (1) пропускає вихідний сигнал на динамік. Таким чином, користувач комп'ютера може задіювати тільки канал 2. Частіше всього його застосовують для генерації звуків заданої частоти і тривалості. Крім того, вихідний сигнал даного каналу програмно доступний по читанню з одного з розрядів паралельного порту. Це дозволяє, запрограмувавши таймер відповідним чином, витримувати потрібні тимчасові інтервали. Для цього слід програмно дозволити генерацію (при відключеному динаміку), а потім програмно опитувати вихідний сигнал таймера і ухвалювати рішення по зміні його рівня.

Підсистема годинника реального часу в перших комп'ютерах виконувалася на мікросхемі контролера MC146818 фірми Motorola. Цей контролер містить 64 байти CMOS-пам'яті, з яких перші 14 байт використовуються для годинника реального часу, а інші 50 байт беруть інформацію про конфігурацію системи. Для вхідного тактового сигналу контролера застосовується спеціальний «годинний» кварцовий генератор з частотою 32 768 кГц, що дозволяє за допомогою розподілу частоти одержати імпульси з частотою 1 Гц. контролер рахує секунди, хвилини, години, дні тижня, місяці і роки. Причому працює він навіть при відключенні живлення комп'ютера, підживляючись від батареї або акумулятора. Це дозволяє зберігати інформацію про поточний час постійно.

Крім лічильника поточного часу, контролер має в своєму складі будильник. Будильник може формувати переривання (IRQ8) з програмно заданою періодичністю. Стани всіх лічильників (секунд, хвилин, годинника і т.д.) програмно доступні як по читанню, так і по запису, що дозволяє встановлювати потрібний час і стежити за ним.

Засоби інтерфейсу користувача

Для зв'язку комп'ютера з користувачем (тобто організації інтерфейсу користувача) застосовуються відеоадаптер, що управляє відеомонітором, клавіатура і графічний маніпулятор типу «миша» (mouse), touch pad або stick pointer. Відеоадаптер є пристроєм сполучення комп'ютера з відеомонітором і частіше всього виконується у вигляді спеціальної плати розширення, що вставляється в системну шину або локальну шину комп'ютера. При цьому зображення, формоване на екрані монітора, зберігається у відеопам'яті, що входить до складу відеоадаптера. Відеопам'ять є оперативною пам'яттю, яка, хоча і не є, по суті, системною пам'яттю, розглядається процесором як частина системної пам'яті з адресами A0000 - BFFFF (всього 128 Кбайт). Тобто з цією пам'яттю процесор може взаємодіяти як з системною оперативною пам'яттю: писати інформацію в будь-яку комірку і читати інформацію з неї. Але одночасно ця ж пам'ять постійно сканується (тобто послідовно опитується) самим відеоадаптером для формування растрового зображення на екрані монітора. Тобто доступ до цієї пам'яті мають як процесор, так і відеоадаптер. Швидкість обміну з відеопам'яттю - досить важливий параметр, він впливає на зручність роботи з комп'ютером і часто визначає коло задач, які можуть ним виконуватися. Тому для відеопам'яті використовують самі швидкодійні мікросхеми. Крім того, застосовують спеціальні архітектурні рішення, що дозволяють полегшити розділення доступу до пам'яті з боку процесора і відеоадаптера. Наприклад, у разі двохпортової пам'яті VRAM - Video RAM, до кожної її комірки одночасно можуть дістати доступ (із записом або читанням) як процесор, так і сам адаптер. Відзначимо, що в старих відеоадаптерах для зниження спотворень зображення на екрані під час перезапису вмісту пам'яті використовувалося звернення до пам'яті з боку центрального процесора тільки в періоди кадрового і рядкового гасячого імпульсів (коли електронний промінь монітора гаситься при переході до наступного рядка екрану або до наступного кадру). Всі сучасні відеоадаптери можуть працювати в двох основних режимах: текстовому (символьному, алфавітно-цифровому) і графічному. В текстовому режимі відеопам'ять має початкову адресу B8000, а в графічному - A0000. В текстовому режимі на екран можна виводити тільки окремі символи, причому тільки в певні позиції на екрані. При цьому у відеопам'яті зберігаються виключно коди символів (8-розрядні) і коди атрибутів символів, що виводяться (8-розрядні). Тобто кожній символній позиції на екрані відповідає два байти пам'яті. До атрибутів символу відносяться яскравість, колір, мерехтіння як символу, так і його фону. Для перетворення вмісту пам'яті у відеосигнал точкового зображення застосовується так званий знакогенератор. Він може бути у вигляді ПЗП, в якому записано порядкове растрове зображення кожного символу. При цьому чим більше точок растру відводиться під зображення символу, тим він якісний, але тим більше місця займає на екрані. Переваги текстового режиму - це простота управління екраном і малий об'єм необхідної пам'яті. Прикладом його використання є програма початкового запуску BIOS.

В графічному режимі у відеопам'яті зберігається опис кожної крапки на екрані монітора. Кожній крапці відповідає декілька біт пам'яті (використовується ряд: 1, 4, 8, 16, 24 біт на одну крапку). При цьому, відповідно, кожна крапка може мати 2^n станів, де n - кількість бітів, а під станом розуміється колір і яскравість крапки. При одному біті крапка може бути білою або чорною, при 4 бітах вона може мати 16 кольорів, при 8 бітах - 256, при 16 бітах - 65 536, а при 24 бітах - 16 777 216 кольорів і відтінків. Тут же відзначимо, що загальна кількість крапок на екрані в сучасних комп'ютерах вибирається з ряду 640 (по горизонталі) x 480 (по вертикалі), 800x600, 1024x768, 1280x1024, 1600x1200. Звідси неважко розрахувати

об'єм відеопам'яті, що вимагається для повного екрану. Так, наприклад, при роздільній здатності 800x600 точок і при 256 кольорах (8 біт або 1 байт) потрібен $800 \times 600 \times 1 = 480\,000$ байт пам'яті. При роздільній здатності 1024x768 і 65 536 кольорів (2 байти) потрібні $1024 \times 768 \times 2 = 1\,572\,864$ байти. Проте об'єм відеопам'яті вибирається з наступного ряду: 256 Кбайт, 512 Кбайт, 1 Мбайт, 2 Мбайт, 4 Мбайт, 8 Мбайт, 16 Мбайт. В табл. 7.2 приведені необхідні об'єми відеопам'яті для різних режимів роботи відеоадаптера.

Таблиця 7.2. Необхідні об'єми відеопам'яті.

Роздільна здатність та кількість кольорів	Кількість біт на точку	об'єм відеопам'яті
800 x 600, 16 цветов	4	256 Кбайт
800 x 600, 256 цветов	8	512 Кбайт
800 x 600, 64 К цветов	16	1 Мбайт
800 x 600, 16 М цветов	24	2 Мбайт
1024 x 768, 16 цветов	4	512 Кбайт
1024 x 768, 256 цветов	8	1 Мбайт
1024 x 768, 64 К цветов	16	2 Мбайт
1024 x 768, 16 М цветов	24	4 Мбайт
1280 x 1024, 16 цветов	4	1 Мбайт
1280 x 1024, 256 цветов	8	2 Мбайт
1280 x 1024, 64 К цветов	16	4 Мбайт
1280 x 1024, 16 М цветов	24	4 Мбайт

Зрозуміло, що для повного оновлення такого великого об'єму пам'яті потрібен значний час навіть при швидкій відеопам'яті і швидкому процесорі. В ролі обмежуючого чинника виступатиме темп обміну по системній шині. Тому саме відеоадаптери першими стали розміщувати на локальній шині VLB або на шині PCI, а пізніше - на виділеній шині AGP. Інший напрям прискорення формування зображення - вдосконалення принципів обміну з комп'ютером. Перші відеоадаптери були розраховані на те, щоб всі маніпуляції із зображенням проводив сам центральний процесор комп'ютера. Принципово інший підхід - використання графічного співпроцесора. При цьому центральний процесор тільки дає команди на формування зображення, а співпроцесор, розташований на платі відеоадаптера, сам вже виконує всю обробку, розрахунки і формування об'єктів на екрані, що дає велике збільшення швидкості формування зображень. Проміжний варіант - це вживання так званих графічних прискорювачів, тобто вузлів, що виконують самі трудомісткі операції по формуванню зображень, але центральний процесор при цьому не звільняється повністю від управління відеопам'яттю.

В даний час найбільш поширено два стандарти дисплеїв: **SVGA (Super VGA)**, який підтримує максимальну роздільну здатність 1024x768 точок (стандартними вважається 800x600 точок) в 16- і 256-кольорових режимах при максимальному об'ємі відеопам'яті 4 Мбайт. Крім того, передбачено використання двохпортової пам'яті і 16-розрядної шини даних і ряд інших новин. **XGA і XGA-2 (eXtended Graphics Array)** - ці стандарти запропоновані в 1990 і 1992 г.г. компанією IBM. Основним режимом вважається роздільну здатність 1024x768 точок при 256 кольорах (XGA) або при 64 К кольорах (XGA-2). Відмітна особливість - використання швидкодійного графічного співпроцесора і наявність можливості управляти системною шиною, що дозволяє виконувати відеооперації без участі центрального процесора. Так само, як і в SVGA, використовується двохпортова оперативна пам'ять, причому вона розташовується в адресному просторі комп'ютера в останніх адресах повної 4-гігабайтної області, на які звичайно ніхто не претендує. В XGA-2, на відміну від XGA, використовується тільки прогресивна (суцільна, non-interlaced, NI), а не чересрядкова (interlaced) розгортка зображення на екрані монітора, що забезпечує малі мерехтіння. Обидва стандарти

підтримують повну сумісність з SVGA.

UVGA (Ultra VGA) – основною роздільною здатністю вважається 1280x1024 точок.

UXGA – роздільна здатність 1600x1200 точок, **XVGA** - 1280x768 крапок.

Для підключення до комп'ютера клавіатури застосовується спеціальний інтерфейс з послідовною передачею інформації. Це дозволяє використовувати для приєднання клавіатури всього два двонаправлені дроти (лінія даних і тактовий сигнал). Обмін інформацією йде 11-бітовими послілками, що включають 8 розрядів даних і службову інформацію (тобто стартовий біт, біт парності і стоповий біт). В комп'ютері IBM PC XT для підключення клавіатури використовувалася мікросхема PPI (Programmable Peripheral Interface) i8255, а в PC AT - мікросхема UPI (Universal Peripheral Interface) i8042. Принцип роботи клавіатури досить простий. Він зводиться до постійного сканування (послідовному опиті) всіх клавіш (звичайно застосовується 101-клавішна клавіатура) і до пересилки в комп'ютер номера натисненої клавіші (8-бітовий код), причому як при її натисненні, так і при відпуску. При відпуску клавіші її код передує послілкою коду F0. Якщо клавіша утримується тривалий час, то через заданий інтервал послілки її код повторюються із заданою частотою. Якщо одночасно натиснено більше однієї клавіші, то повторюється послілка коду тільки останньої з натиснутих клавіш. При отриманні коду контролером 8042 він формує сигнал запиту апаратного переривання IRQ1. Це приводить до виклику програми обробки натиснення клавіші, що знаходиться в BIOS. Службові клавіші (Shift, Ctrl, Alt) і перемикаючі клавіші (Caps Lock, Insert, Num Lock) обробляються спеціальним чином, а у разі натиснення символічних клавіш їх коди перетворюються в коди відповідних символів і поміщаються в буфер клавіатури. Буфер клавіатури - це 16-байта область пам'яті, організована за принципом FIFO «перший ввійшов - перший вийшов», в якій зберігаються коди натиснутих клавіш до тих пір, поки їх зможе обробити програма.

Сучасні клавіатури персональних комп'ютерів мають 101 або 102 клавіші. Є «розширені» моделі з кількістю клавіш до 122 і «усічені» моделі з кількістю клавіш близько 90, вживані в портативних комп'ютерах типу ноутбук.

Починаючи з комп'ютера PC AT, клавіатура може не тільки передавати інформацію, але і приймати її. Ця можливість використовується для пересилки в клавіатуру команд, що встановлюють режими її роботи (наприклад, швидкість повтору введення символів при утримуванні клавіші або тимчасова затримка перед повтором). Комп'ютерна миша, що служить для управління курсором, підключається до комп'ютера через стандартний послідовний інтерфейс RS-232C (про нього докладніше - в окремому розділі). Для передачі комп'ютеру інформації про переміщення миші використовується 3-байтовий формат. Два байти при цьому містять інформацію про переміщення миші по вертикалі і по горизонталі, а один байт - про стан кнопок миші. Передача ведеться тільки в одному напрямі (від миші до комп'ютера) із швидкістю 1200 бит/с. Переміщення вимірюється в спеціальних одиницях срі (counts per inch), рівних приблизно 0,005 дюйма (0,13 мм).

Варто відзначити, що миша, як правило, живиться від системного блоку комп'ютера, для чого задіяні неживані сигнальні лінії роз'єму інтерфейсу RS-232C, оскільки власне напруги живлення на роз'єм не виведені. Саме тому миша приєднується до комп'ютера чотирьохдротним кабелем, хоча для інформації вистачило б і двохдротного. Докладніше про інтерфейс RS-232C в наступному розділі. Зараз використовується також підключення миші через інтерфейс PS/2, схожий на RS-232C, але не сумісний з ним ні електрично, ні конструктивно.

Альтернатива миші - це маніпулятори Stick Pointer і Touch Pad, які не мають механічних частин, що рухаються. Спочатку вони застосовувалися тільки в ноутбуках, але потім їх стали розміщувати і на клавіатурах звичайних настільних комп'ютерів. Stick Pointer є невеликим важелем, розташованим між клавішами. Тиск на нього в різні боки викликає переміщення курсора на екрані. При цьому сам важіль залишається нерухомим. Touch Pad є невеликим майданчиком, розташованим поряд з клавішами, по якій необхідно рухати пальцем або ручкою, причому рух пальця викликає таке ж переміщення курсора на екрані. З погляду

комп'ютера ці маніпулятори нічим не відрізняються від миші, вони використовують той же інтерфейс.

Ігровий адаптер джойстик підключається до комп'ютера через власний спеціальний інтерфейс. Для зв'язку з джойстиком не вимагається ніяких переривань. Використовується тільки одна адреса вводу/виводу.

Зовнішня пам'ять

Зовнішня пам'ять комп'ютера є дисковими накопичувачами інформації - вбудований накопичувач на жорсткому диску (вінчестер) і накопичувач на змінних гнучких дисках (дискетах). В обох випадках магнітні диски бережуть інформацію у вигляді намагнічених концентричних доріжок (циліндрів) на магнітному покритті, розбитих на сектори. Диск в накопичувачі постійно обертається, а запис і читання інформації проводяться переміщуваними уздовж радіусу диска магнітними головками. Завдяки постійному прогресу технології виробництва накопичувачів, розвитку технології магнітних покриттів і магнітних головок, ємність вінчестерів підвищилася до декількох десятків гігабайт, а ємність дискет - до сотень мегабайт (правда, стандартним поки вважається об'єм дискети 1,44 Мбайт). Докладний опис роботи дисководів і принципів зберігання інформації на магнітних дисках зажадав би дуже багато місця, до того ж воно не має прямого відношення до теми даної книги, тому ми тут приведемо тільки деякі особливості організації обміну інформацією. Важливий параметр будь-якого дисководу - ця його швидкодія, яка визначається, з одного боку, досяжною швидкістю запису/читання інформації, а з іншою - часом позиціонування (тобто установка в потрібне положення) магнітної головки дисководу. Важлива і швидкодія інтерфейсу, що здійснює зв'язок комп'ютера з накопичувачем, а також вживані способи організації обміну інформацією.

В даний час найбільш поширено два стандартні інтерфейси для вінчестерів:

- IDE (Integrated Drive Electronics) - інтерфейс для дискових накопичувачів, офіційна назва - ATA (AT Attachment). Саме цей інтерфейс застосовується як основний в персональних комп'ютерах. Швидкість обміну може досягати 133 Мбайт/с.
- SCSI (Small Computer System Interface) - малий комп'ютерний системний інтерфейс. У принципі, він використовується і для підключення інших пристроїв (наприклад, сканерів), але основне його вживання - для дисководів. Як правило, даний інтерфейс спочатку включається в структуру тільки деяких серверів, а для його реалізації в персональних комп'ютерах необхідна додаткова плата розширення (до речі, досить дорога). Швидкість обміну може досягати 320 Мбайт/с.

Порівняння цих двох інтерфейсів (SCSI і IDE) показує, що в розрахованих на одного користувача автономних системах набагато ефективно застосовувати IDE, а в розрахованих на багато користувачів і багатозадачних системах вигідніше стає SCSI. Варто також відзначити, що установка SCSI складніше і дорожче, ніж IDE. Крім того, при використуванні вінчестера з інтерфейсом SCSI як мережний диск можуть виникнути проблеми. Перевагою SCSI є більша кількість максимально підключених дисководів і можливість одночасного виконання ними команд, що подаються. А що стосується швидкості обміну, то вона в основному визначається не пропускною спроможністю інтерфейсу, а іншими параметрами, зокрема швидкістю системної шини, що використовується. Тому точно сказати, дисковод з яким інтерфейсом працюватиме швидше, в загальному випадку неможливо. До того ж у разі IDE реальна швидкість дуже сильно залежить від рішень схемотехнік, використаних виготівником дисководів.

Для прискорення обміну з дисками широко застосовується кешування, принцип якого близький до принципу кешування оперативної пам'яті. Так само кешування диска дозволяє за рахунок використання більш швидкої електронної пам'яті, ніж дискова пам'ять, істотно збільшити середню швидкість обміну з диском. Тут принципово важливо декілька моментів:

- в більшості випадків кожне наступне звернення до диска буде зверненням до наступного по порядку блоку інформації на диску;
- для позиціонування головки потрібен помітний час (порядка мілісекунди);
- шуканий сектор на диску може не виявитися під головкою після її установки, і буде потрібно чекати його приходу.

Все це призводить до того, що виявляється набагато вигідніше містити в оперативній пам'яті (дислової кеш-пам'яті) копію частини диска і звертатися на диск тільки в тому випадку, якщо потрібної інформації немає в кеш-пам'яті. Для обміну з кеш-пам'яттю, як і у разі оперативної пам'яті, використовуються методи Write Through (WT) і Write Back (WB). Оскільки вінчестер - це блоково-орієнтований пристрій (розмір блоку рівний 512 байт), то дані передаються в кеш блоками. При заповненні кеш-пам'яті в неї переписуються не тільки необхідні в даний момент блоки, але і наступні за ними (метод «читання вперед», Read Ahead), подальше звернення до яких найбільш вірогідне. Особливо ефективно кешування при оптимізації жорсткого диска (його дефрагментації), коли кожний файл розташований в групі секторів, наступних один за одним. Як і у разі кешування пам'яті, при кешуванні диска використовується механізм LRU, що дозволяє обновляти ті блоки, до яких довше за все не було обігу. Кеш-пам'ять диска звичайно розташовується на платі спеціального кеш-контролера дисководу, і її об'єм може досягати 16 Мбайт. Для сполучення з комп'ютером дисководу для гнучких дисків (флоппи-дисків, дискет) традиційно застосовується спеціальний інтерфейс SA-400, розроблений на початку 70-х років. контролер приєднується до дисководу 34-дротяним кабелем, причому до одного контролера звичайно приєднується до двох дисководів (теоретично їх може бути чотири). На кожному накопичувачі, як правило, є чотири перемички DS0-DS3 (Drive Select) для вибору номера даного дисководу. Дані по інтерфейсу передаються в послідовному коді в обох напрямках (по різних дротах). Швидкість передачі даних для дискет місткістю 1,44 Мбайт складає 500 Кбіт/с. Як і контролер жорстких дисків, контролер гнучких дисків в сучасних комп'ютерах встановлений на системній платі (для старих моделей комп'ютерів випускалася спеціальна плата розширення). В нових комп'ютерах став стандартним дисковод на оптичних компакт-дисках (CD-ROM). На цих дисках інформація зберігається у вигляді зон з різними ступенями віддзеркалення світла від поверхні диска. Замість безлічі концентричних доріжок на поверхні диска (як у магнітного диска, вінчестера), у разі компакт-диска застосовується всього одна спіральна доріжка. Для читання інформації застосовується мініатюрний лазер. Диски мають діаметр 5 дюймів і стандартний об'єм 780 Мбайт. Швидкість обміну інформацією з компакт-дисками зараз складає від 2,4 Мбайт/с (для дисководів із швидкістю 16x) до 3,6 Мбайт/с (для дисководів із швидкістю 52x). Використовуються інтерфейси IDE і SCSI. На компакт-диск записуються не тільки дані, але і звук, а також зображення. Існують компакт-диски з можливістю однократного запису або навіть багаторазового перезапису інформації з комп'ютера. Можливо, дисководи, що підтримують такі диски, незабаром увійдуть до стандартної комплектації персонального комп'ютера. Правда, швидкість запису інформації на компакт-диски звичайно істотно нижче за швидкість читання інформації.

Зразу ж обмовимося, що під інтерфейсами персонального комп'ютера в даному випадку маються на увазі тільки зовнішні інтерфейси, тобто засоби сполучення із зовнішніми по відношенню до комп'ютера в цілому пристроями. При цьому зовнішні пристрої можуть бути як стандартними (наприклад, принтер або модем), так і нестандартними (наприклад, вимірювальні і управляючі модулі, прилади, установки). В даний час комп'ютери можуть мати безліч зовнішніх інтерфейсів. Найбільш поширені наступні:

- системна шина (магістраль) ISA;
- шина PCI;
- шина AGP;

- шина PC Cards (стара назва PCMCIA) - звичайно тільки в ноутбуках;
- паралельний порт (принтер, LPT-порт) Centronics;
- послідовний порт (COM-порт) RS-232C;
- послідовний порт USB (Universal Serial Bus);
- послідовний інфрачервоний порт IrDA.

Крім того, комп'ютери можуть мати роз'єми для підключення зовнішнього монітора, клавіатури, миші. Деякі комп'ютери мають вбудовані модеми і мережні адаптери, тоді вони розташовують, відповідно, телефонним і мережним зовнішніми інтерфейсами. Підключення стандартних зовнішніх пристроїв звичайно не викликає жодних проблем: треба тільки приєднати пристрій до комп'ютера відповідним стандартним кабелем і (можливо) встановити на комп'ютер програмний драйвер. Знати особливості зовнішніх інтерфейсів користувачу в даному випадку не обов'язково. У разі інфрачервоного порту не потрібен навіть кабель.

Набагато складніше ситуація, коли до комп'ютера вимагається приєднати нестандартний зовнішній пристрій. В цьому випадку необхідне доскональне знання особливостей інтерфейсів, що використовуються, і уміння ефективно з ними працювати. Обмежений об'єм книги не дозволяє повністю розглянути дане питання, тому ми зупинимося тільки на загальному описі деяких зовнішніх інтерфейсів комп'ютера.

Частіше за все для підключення нестандартних зовнішніх пристроїв використовуються системна магістраль ISA, паралельний інтерфейс Centronics (LPT) і послідовний інтерфейс RS-232C (COM).